Personality Prediction using Machine Learning

Mushfika Rahman

May 10,2022

Abstract

This project aims to build a machine learning-based classifier that can predict personality into Myres-Briggs Type Index (MBTI). The model was built using data collected through social media posts. There has been significant research on predicting and recognizing patterns of human psychology. Since the inception of machine learning is learning and predicting patterns from data, personality prediction would be an interesting application of machine learning. Moreover, the project aimed to find the parameters that affect the learning of text-based data. The project focused on finding a learning algorithm for building a predictive model that would provide good accuracy and generalize well.

1 Introduction

Observing and predicting human personality has been researching interest in behavioral learning. The Myers–Briggs Type Indicator (MBTI) is one of the most popular and reliable methods for predicting personality. The Myers Briggs Type Indicator is a personality type system that divides everyone into sixteen distinct personality types across four axes: Introversion– Extroversion, Intuition – Sensing, Thinking – Feeling, Judging – Perceiving. Combining the four-axis variables results in the final personality type. For instance, a person with ENTJ personality is a combination of extraverted, intuitive, thinking, and judging [3]. Several technological advancements over the last decade have enabled researchers to devise various new methods for collecting data in personality science. Moreover, with the increased use of smartphones and social media, collecting data about human behavior has been relatively easy. Along with the improvement in the collection and availability of such data, significant advancements have been made in building methods that can be used to model these complex data. With the availability of massive multidimensional data about human behavior, research has shown it is possible to use machine learning to predict human personality better than humans [4].

Many studies have been conducted using machine learning on various data types to determine individuals' personality traits. This research area is popular because of its practical applications in modern technology. In a targeted marketing campaign, personality prediction is used. Significant research has been conducted on incorporating personality into the development of a recommendation system. On the other hand, supervised algorithms are used in applications such as Naive Bayes and Bagged Support Vector Machine [2]. Researchers are currently experimenting with unsupervised algorithms to predict personality. A significant amount of research effort is devoted to improving the model's accuracy and running time.

The project focused on efficiently implementing supervised machine learning algorithms into textual data. The choice of supervised learning algorithms was *Logistic Regression* and *Support Vector Machine*(SVM) for the project. Logistic regression is a simple and more efficient method for binary and classification problems. Logistic regression is a method for estimating the likelihood of an outcome. The project is trying to predict personality across the different axis; a probabilistic result will be a good choice([1]). However, logistic regression is a binary class classifier, and our prediction is a multiclass classifier; the project utilized a multinomial logistic regression classifier. On the other hand, SVM is very efficient while working with high-dimensional space. Furthermore, previous research on text-classification problems showed that SVM could generalize more accurately on unseen cases relative to classifiers([1]). Because of these properties, SVM was selected for predicting purposes.

Firstly, the project implemented a multiclass classifier for predicting 16 personality types using multinomial logistic regression and SVM. SVM is a binary classifier; the project used the OneVsRest approach to convert it into a multiclass problem. In both cases, they generalized poorly. The reason behind that, the problem was defined to predict personality on four different axes. Thus, the problem was broken down to classifying four different binary classifiers; Introversion– Extroversion, Intuition − Sensing, Thinking − Feeling, Judging − Perceiving. The algorithm's performance increased significantly on the classifiers. The project experimented with the impact of the algorithm's accuracy on changing parameters and hyperparameters. The project experimented with manipulating no data points, no features, and regularization, kernel choice (SVM) to achieve accuracy. The outcome for Introversion– Extroversion (81%), Intuition − Sensing(84%), Thinking − Feeling(80%), Judging − Perceiving(72%) in multinomial logistic regression.

2 Background Study

Youyou et al.[4] researched the accuracy between human and computer-based personality judgments. The experiment was on the OCEAN model's five traits of (Openness, Agreeableness, Extroversion, Conscientiousness, and Neuroticism). The result showed that the computer's average was more remarkable than the Human's average accuracy [4]. The result indicates that Machine learning models are particularly adapted to these types of data, allowing researchers to model very complicated relationships and use feature extraction approaches to assess the generalizability and robustness of their findings. One of the earliest research from Wang et al. [9] showed the success of personality prediction on MBTI using logistic regression. Thus, I was inspired to experiment with the algorithm.

Pratama et al. [7] build a personality classifier from social media data on OCEAN's model. The authors experimented with different algorithms in English data sets and Indonesian data. Additionally, they build classifiers on different learning algorithms, Naive Bayes, K-Nearest Neighbors, and Support Vector Machine. Furthermore, they build a classifier by combining all the classifiers' outputs. Their proposed approach outperformed other algorithms by around 65%. [7]. Their result indicated that all the classifiers performed closely, where Naive Bayes surpassed slightly. However, the work by Tandera et al. [8] showed that SVM performed better. The authors showed a comprehensive analysis of different algorithms through accuracy results. The authors experimented with logistic regression, Naive Bayes, SVM, Logistic Regression, and deep neural network architecture, and the neural network achieved around 72% accuracy [8].

Ninda et al. performed a study on classifying the personality of Facebook users into one of the Big Five Personality Traits using SVM. Their result provided the best accuracy of 87.5% from SVM [1]. The proposed project wanted to explore old machine learning algorithms, and the majority of the research showed success in Support Vector Machine(SVM) therefore inspired to implement SVM.

3 Dataset

The data set was collected from Kaggle[5]. It consists of personality types and social posts of around 8500 people. It is a labeled dataset that consists of 16 types of distinct human personalities. Each user has their latest 50 social media posts. The dataset was augmented to include the four axes of the MBIT index. Introversion(labeled as 0 class)–Extroversion(labeled as 1 class),Intuition(labeled as 0 class) − Sensing(labeled as 1 class), Thinking(labeled as 0 class) − Feeling(labeled as 1 class), Judging(labeled as 0 class)

– Perceiving(labeled as 1 class).

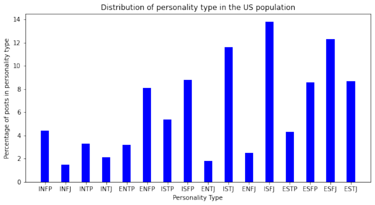

Figure 1: frequency percentage of personality type among USA population

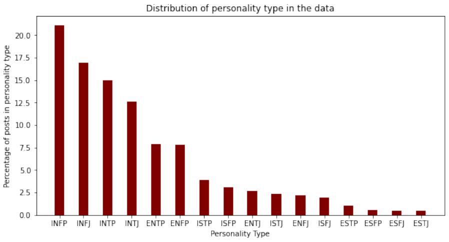

Figure 2: frequency of personality type in dataset

The dataset was disproportional across the different classes. One of the exciting aspects of the dataset distribution is that it did not adhere to the distribution in the general population; the most popular type of personalities such as ESTP (10%), ESFP (11%), ESFJ (12%), and ESTJ (13%) [6], were least represented in the collected dataset.

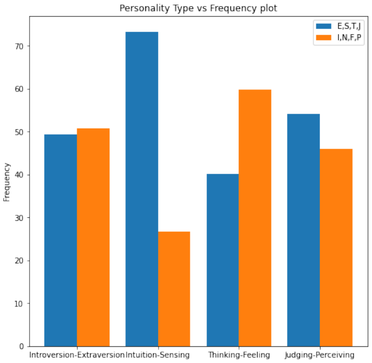

Figure 3: frequency percentage of 4-axis of personality type among USA population

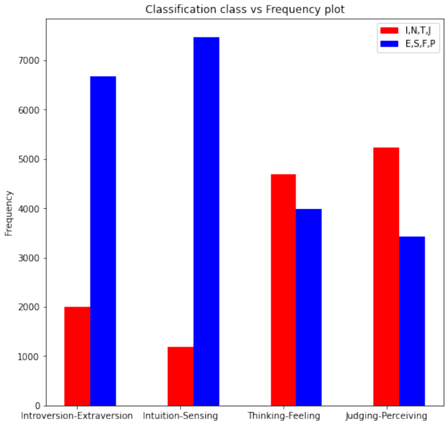

Figure 4: frequency of 4-axis of personality type in dataset

4 Methodology

The project implemented an approach mixture of build-and-fix along with ablative analysis. The data was collected from social media; it was text-based data. Machine learning models work with numeric data. Therefore it is necessary to process the data before the training process. The pre-processing step was to split the dataset into train and test sets (80-20 ratio) to evaluate the performance. The learning algorithm was applied to the training set, experimenting with increasing features, data points, and hyper-parameter tuning gradually increased the performance.

4.1 Pre-Processing

4.1.1 Selected Word Removal

It is mentioned that data was collected from social media; it contained unnecessary words. The data contained different URL links, our model needed to perform well in English languages, and URL links do not contribute to that. It was necessary to remove them. Additionally, all the words were converted into the lower case after removing the URL links. Some words are very commonly used, but they carry very little helpful information; these words are referred to as stopwords. Therefore, it is necessary to remove them. I used python's NLTK library for that. The dataset contained the personality indicator words such as 'entj','inpf', it can trick the model into learning these words, which will not generalize well, so they also need to be removed.

4.1.2 Lemmatization

It is necessary to normalize text because it is essential to reduce its randomness. Furthermore, efficiency is improved by reducing the amount of different information that the computer has to deal with. Two techniques are stemming and lemmatizing. Stemming algorithms work by taking a list of common prefixes and suffixes and cutting off the end or beginning of the word. However, Lemmatization considers the linguistic analysis of the words; therefore, this technique is applied with python's NLTK library.

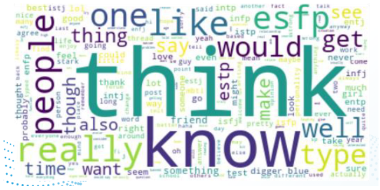

Figure 5: Most frequent words with some pre-processing in ESFP

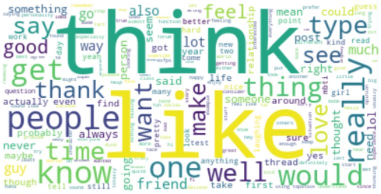

Figure 6: Most frequent words after pre-processing ESFP

4.1.3 Vectorization & Feature Extraction

Term frequency-inverse document frequency(TF-IDF) is a text vectorizer that converts text into a vector that can be used for training purposes. TF-IDF also provides analyzation of the relevancy of words across a corpus in a corpus collection and provides the importance of words in data. TF-IDF was also used for vectorization as well as feature selection. The dataset had around 24848 features for each user post which was trimmed down to smaller feature numbers because the dataset only had 8500 data points. I experimented with various number of feature sets.

5 Experiments

I implemented the multinomial logistic regression to predict 16 types of personality. Firstly, I experimented with the variable number of features.

Accuracy	500 features(Multinomial)	1000 features(Multinomial)
Training Set 787	100%	100%
Testing Set	39%	33%

I experimented with 500 features, then I increased to 1000 features accuracy decreased for multinomial logistic regression, but the learning algorithm provided 0% error in both cases. This inferred that the models are overfitting, thus applying regularization, which trivially increased performance. I also implemented different kernels and applied SVM (OneVsRest); in every case, the training set error was close to zero, and the testing set error was close to 60-50%. One interesting result was achieved while experimenting with SVM; in the polynomial choice of the kernel with degree 3, the model achieved around 72% accuracy on the testing set.

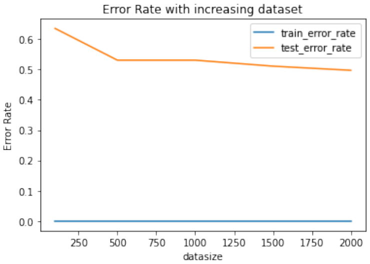

Figure 7: Performance of multinomial logistic regression without regularization

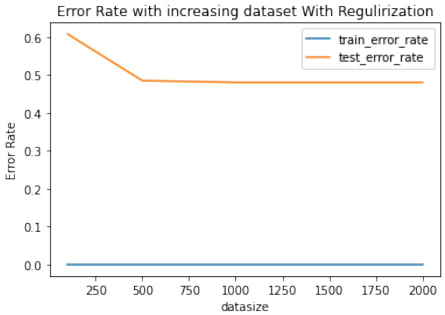

Figure 8: Performance of multinomial logistic regression with regularization

The implementation of four binary classifiers for each axis improved the performance of the algorithms. Multiple runs of the experiment produced the final outcome. Firstly, I experimented with several features to determine the effectiveness of increasing features. In both cases, increasing the feature numbers worked well till it started to over-fitting occurred. The optimal number of features(most frequently used words) was 1000 in this case.

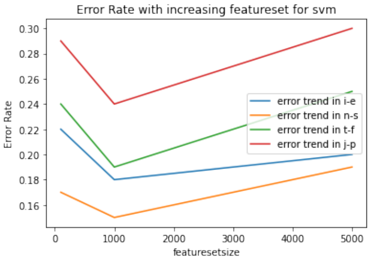

Figure 9: SVM error with increasing featureset

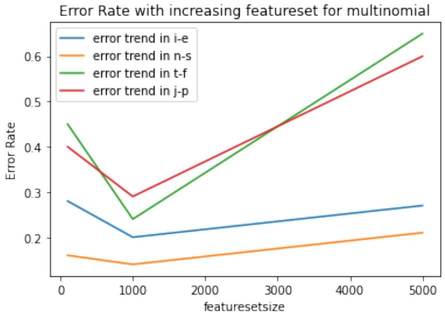

Figure 10: Logistic Regression error with increasing number of features

Then I experimented with the variable number of data points. The algorithm's accuracy increased with more significant data points in both cases. One interesting aspect revealed during this experimentation is that the SVM variance decreases with data points while multinomial logistic variance increases. One example of the variance for Introversion (I)–Extroversion (E) classifier is included in the figure. In SVM, I applied a linear kernel with a soft margin slight bias was observed. and gap between train and the test estimation was closing.

The choice kernel and the margin were selected through grid search cv. I experimented with RBF, linear, polynomial kernel, degree, C, and Gamma. The best estimators returned a linear kernel with C=10.0. I also experimented with RBF-kernel and gamma parameters, and the lower the gamma value was better the accuracy. The multinomial logistic regression across the Intuition(N)-Sensing(S) class performed better than

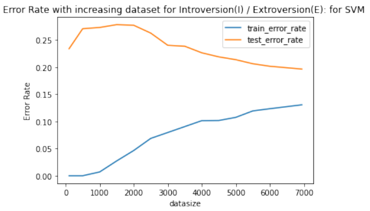

Figure 11: SVM error with increasing datapoints

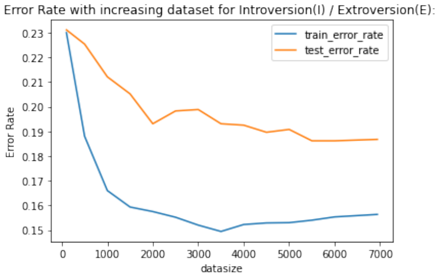

Figure 12: Logistic Regression error with increasing number of datapoints

SVM with linear kernel. On the other hand, in other classifiers, both algorithms performed similarly.

Classifiers	Multinomial	SVM(Linear Kernel, C=10.0)
Introversion (I)–Extroversion (E)	82%	81%
Intuition (N)–Sensing (S)	86%	84%
Thinking (T) – Feeling (F)	80%	80%
Judging (J) –Perceiving (P)	72%	71%

Since the dataset was imbalanced across the classifiers, thus the prediction percentage was not equal throughout every class.

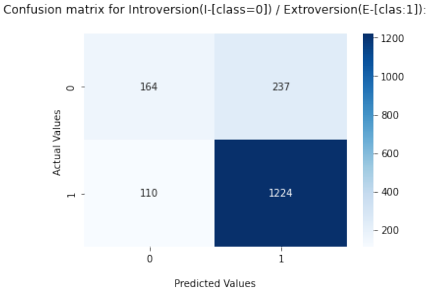

Figure 13: Confusion matrix I-E class in SVM

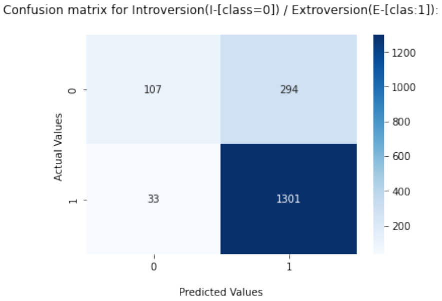

Figure 14: Confusion matrix I-E class in Multinomial Logistic Regression

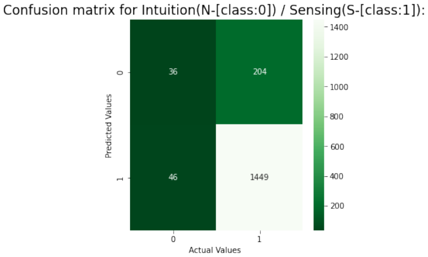

Figure 15: Confusion matrix N-S class in SVM

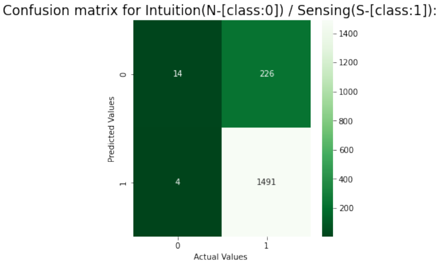

Figure 16: Confusion matrix N-S class in Multinomial Logistic Regression

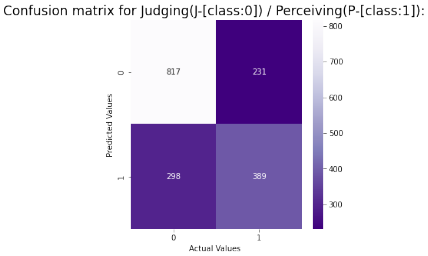

Figure 17: Confusion matrix J-P class in SVM

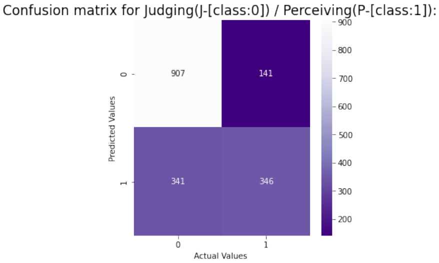

Figure 18: Confusion matrix J-P class in Multinomial Logistic Regression

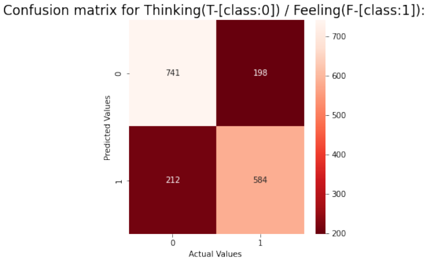

Figure 19: Confusion matrix T-F class in SVM

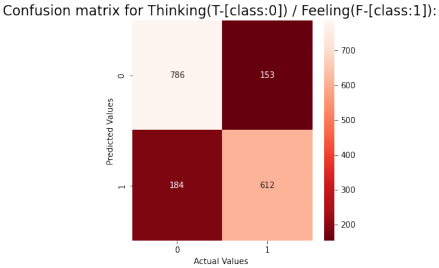

Figure 20: Confusion matrix T-F class in Multinomial Logistic Regression

6 Discussion & Conclusion

The project focused on finding the influence of data points on accuracy. The assumption was that getting more data points would increase the algorithm's accuracy, and the assumption held. Furthermore, I assumed that getting more features would increase the algorithm's accuracy, which did not hold in all cases. This is because with increasing features, the algorithm would pick up some features that do not affect the prediction. In order to tackle the complexity of a more extensive feature set, no of observations should increase. Moreover, the project aimed to find an algorithm that better fits the data. The experiment showed that both the algorithm performed similarly across the classifiers. The project also observed accuracy depends on hyper-parameter tuning. The SVM with a linear kernel was the best to fit data, and the RBF kernel with a lower gamma value produced results close to the linear kernel fit. That may be because the data had some linearity among the features, which could be a matter of further investigation.

Finally, the project wanted to explore if the trained classifiers would produce a similar result to other text-based personality classifications. The assumption is that the trained classifier will not generalize well because the data was trained on data collected from a personality forum, meaning curated for just the MBIT indicator. Moreover, the actual world distribution of the personality type varies from the data set. However, this claim needs further investigation. Additionally, future work also includes improving the performance of the classifiers.

References

[1] `NindaAnggoroUtami,WarihMaharani,ImeldaAtastina,PersonalityClassificationofFacebookUsersAccordingtoBi`
`Method,ProcediaComputerScience,`

[2] Tandera,Tommy&Hendro,&Suhartono,Derwin&Wongso,Rini&Prasetio,Yen.(2017)
.PersonalityPredictionSystemfromFacebookUsers.ProcediaComputerScience.116.604-611.
10.1016/j.procs.2017.10.016.

[3] Myers,IsabelBriggs."TheMyers-BriggsTypeIndicator:Manual(1962)."(1962).

[4] Youyou,Wu,MichalKosinski,andDavidStillwell."Computer-basedpersonalityjudgmentsaremoreaccuratethantho
"ProceedingsoftheNationalAcademyofSciences112.4(2015):1036-1040.

[5] https://www.kaggle.com/datasets/datasnaek/mbti-type

[6] https://www.myersbriggs.org/my-mbti-personality-type/my-mbti-results/
how-frequent-is-my-type.htm

[7] Pratama,BayuYudha,andRiyanartoSarno."PersonalityclassificationbasedonTwittertextusingNaiveBayes,
KNNandSVM."2015InternationalConferenceonDataandSoftwareEngineering(ICoDSE).IEEE,2015.

[8] Tandera,Tommy,etal."Personalitypredictionsystemfromfacebookusers.
"Procediacomputerscience116(2017):604-611.

[9] Wang,Yilun."Understandingpersonalitythroughsocialmedia."JournalofPsychology(2015).