Final Project Data1030

Junhui Huang

December 15 2024

GitHub Repository: `https://github.com/huang-0505/data1030-project`

1 Introduction

1.1 Background information

The Myers–Briggs Type Indicator (MBTI) is a tool designed to categorize individuals into 16 distinct personality types. It is based on four pairs of functional styles: Extroverts, who gain energy from social interaction, and Introverts, who recharge through introspection; Sensing types, who focus on practical details, and Intuitive types, who look for patterns and broader possibilities. Thinkers prioritize logic in decision-making, while feelers rely on empathy and values. Judging types prefer structure and plans, while perceives thrive in flexibility and spontaneity. These pairs combine to offer insight into how people interact, process information, and make decisions.

1.2 Motivation and Data Overview

I think it's always worthy to understand ourselves better and have fun guessing your friends' personality type. Thus, in my final project, I aim to predict individual's personality based on basic features including age, gender, interest, education level, and also features like introversion scores, judging scores, sensing score and thinking scores. My dataset consists a total of 128,061 rows and 9 columns, with no missing values.

1.3 Previous Work

This is a multi-class classification problem. Since the dataset was sourced from Kaggle, many others have attempted to solve the same problem of predicting personality types. In my final predictions, I tested both accuracy and macro f1_score, achieved an accuracy of approximately 90.6% and a macro f1_score of 0.886, which is comparable to the results reported by other contributors on the platform. Even though I will be using macro f1_score as my main metrics, the close accuracy result indicates" that my model is performing on par with established benchmarks for this dataset, reflecting the effectiveness of the preprocessing steps, and model selection employed in my approach.

2 EDA

2.1 Target variable

In figure 1, I initially explored the distribution of my target variable, which is the personality type. It is clear that the dataset is quite imbalanced where ENFP has roughly 35000 counts and ISTJ has only around 1000 counts. It shows that our data is imbalanced, and required stratified splitting.

2.2 Correlation plot between four scores

In Figure 2, I examined the correlations between all continuous features. The diagonal plots reveal that both the introversion and thinking scores follow a uniform distribution. This suggests that the dataset contains roughly equal numbers of introverted and extroverted individuals. The same observation applies to thinking scores.

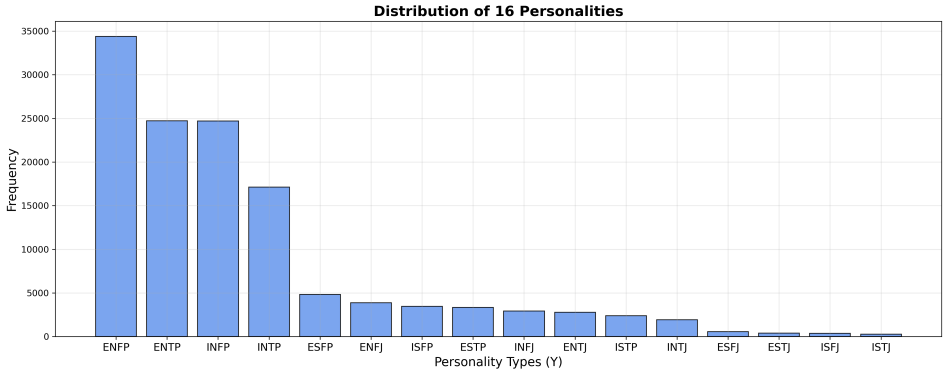

Figure 1: Distribution of 16 personality types.

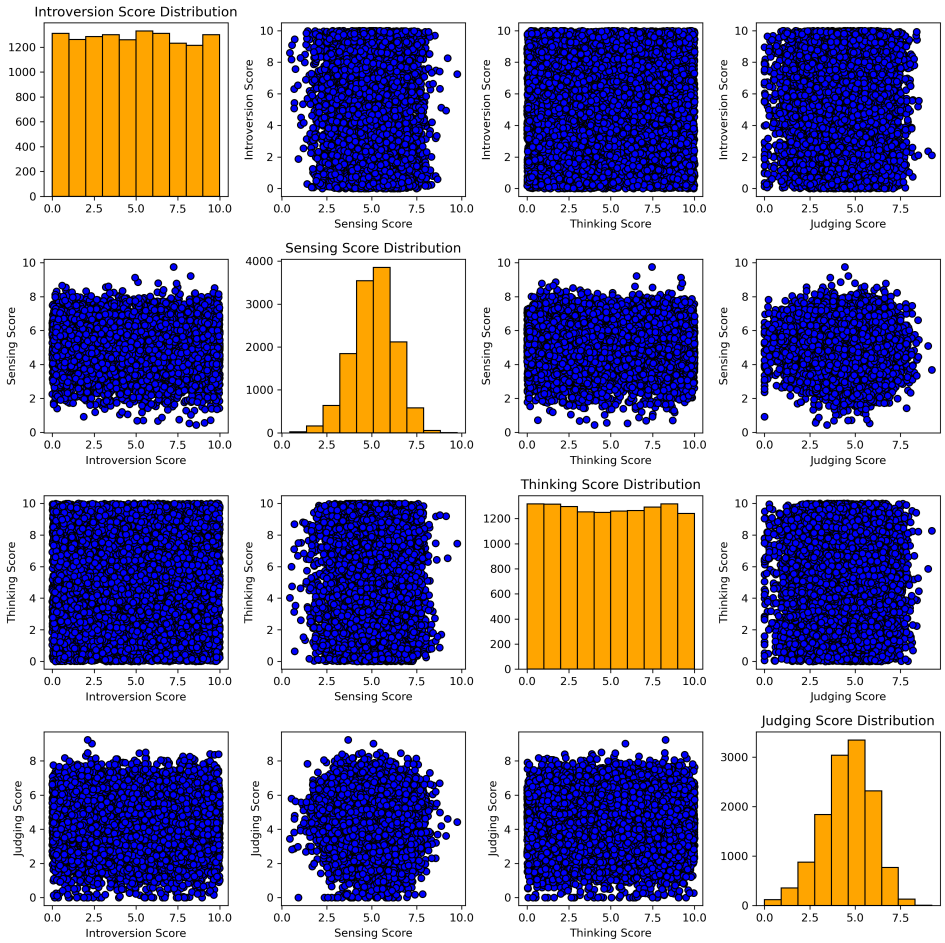

Figure 2: Distribution of interest across all personality types and frequency of each personality type,

Similarly, the sensing and judging scores exhibit normal distributions, indicating that most individuals score near the average for these features, with only a small number of people scoring at the extremes.

2.3 Deeper look into feature of interest

In Figure 3, I analyzed the distribution of interests across different personality types. For most personality types, the largest proportion of individuals falls into the "Unknown" interest category, ranging from 30% to 40%. This suggests that a significant portion of the data lacks recorded or

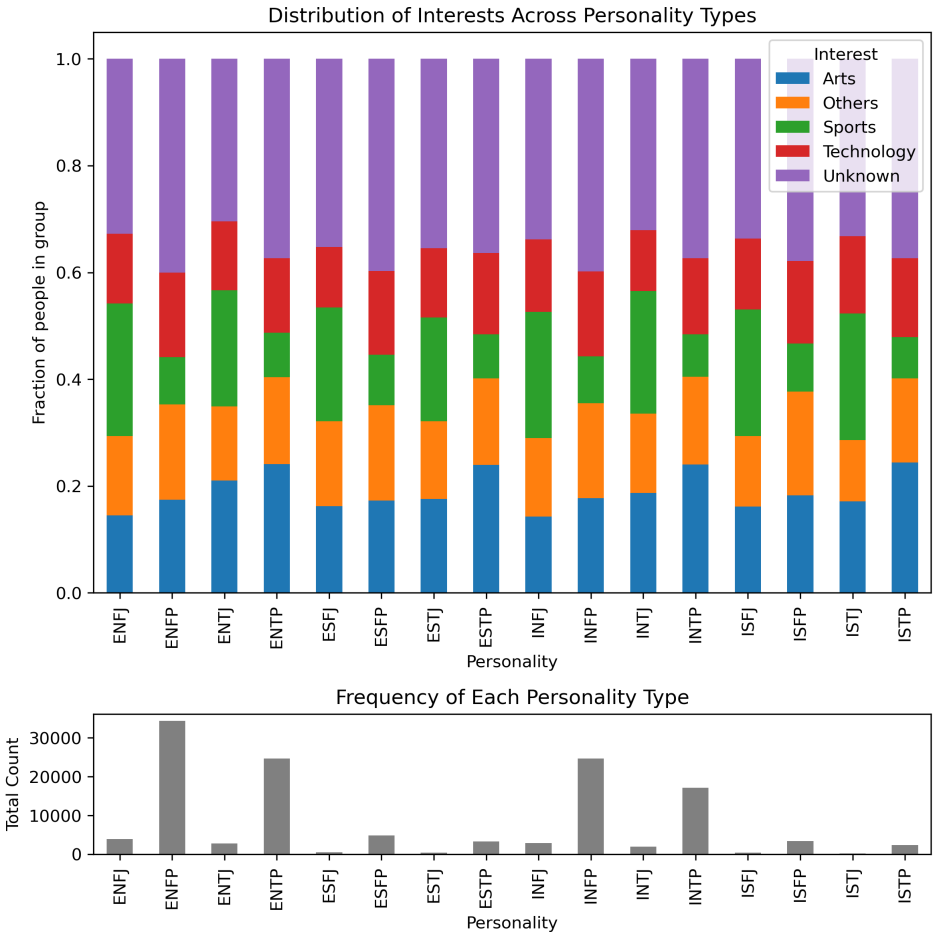

Figure 3: Distribution of Interests Across Personality Types.

categorized interest information.

Grouping the MBTI types by their first two letters (e.g., ENFJ, ENTP, ENTJ), I focused specifically on the Arts category. Within this group, there is a clear upward trend, with the percentage of individuals interested in Arts increasing from ENFJ to ENTP, where ENTP has the highest representation. This trend extends to other groups as well, showing that personality types ending with TP are more likely to have an interest in Arts.

3 Methods

3.1 Data Splitting

I used a two-step data splitting strategy. First, I performed a (90/10) train-test split to separate a 10% test set for final evaluation with stratify because it's a bit imbalanced. Then, I applied 5-fold cross-validation on the remaining 90% (training + validation set) to optimize hyper-parameters and ensure robust model evaluation. Even though I have a pretty large dataset (¿100k), but I think using k-fold validation rather than a simple train-val-test split would be better to optimize available data for model performance.

3.2 Data Prepossessing

I have no missing data in my dataset, thus the data precessing step is mainly encoding and scaling. My dataset comprises 8 features: 1 ordinal feature (Education), 2 categorical features (Interest and Gender), and 5 continuous features (Age, Introversion Score, Sensing Score, Thinking Score, and Judging Score). Accordingly, I applied ordinal encoding to Education, one-hot encoding to Interest and

Gender, Min-Max scaling to Age, and standard scaling to Introversion Score, Sensing Score, Thinking Score, and Judging Score.

3.3 ML Pipeline

My machine learning pipeline contains data prepossessing, model training with hyper parameter tuning using cross validation, and model evaluation. Data prepossessing was discussed in last paragraph, I used gridsearchCV package from sklearn to perform parameter grid search with 5-fold cross-validation on my training and validation set, evaluating combinations of hyperparameters to find the best parameter pairs based on the validation macro f1 score. Once the best model is identified, it is evaluated on the test set to provide an unbiased estimate of its performance. All of these were done in three different random states to evaluate mean and standard deviation to have more accurate result and to measure the uncertainty.

3.4 Metric

I chose to use the macro F1 score because my data is somewhat imbalanced across certain classes. While accuracy is a commonly used metric, it can be misleading in imbalanced datasets. Since all personality classes should be treated equally, the macro F1 score is a better fit. It evaluates performance for each class individually, calculates the F1 score for each, and then uses an unweighted average to ensure equal importance for all classes. Additionally, the macro F1 score balances precision and recall, ensuring both false positives and false negatives are taken into account.

3.5 Description of 5 ML models

I utilized 5 machine learning algorithms, including random forest, support vector machine, XGBoost, Logistic regression and Decision tree. Since my data is imbalanced in some classes, thus I calculated class weight as the inverse of frequency for each class and applied it in all 5 ML algorithms.

Machine Learning Model	**Parameters Tuned and Values Used**
Random Forest	max_depth: {10, **20**, 30}; max_features: {**0.25**, 0.5, 0.75, 1.0}
Support Vector Classifier	C: {0.1, 1, **10**}; gamma: {**scale**, auto}
XGBoost	learning_rate: {0.01, **0.1**, 0.2}; max_depth: {3, **5**, 7}; sub_sample: {0.2, 0.5, **0.8** };
Logistic Regression	C: {0.01, 0.1, **1**}; penalty: {**l1**, l2}
Decision Tree	criterion: {gini, **entropy**}; max_depth: {None, 10, **20**, 30}; min_samples_split: {**2**, 5, 10}; min_samples_leaf: {**1**, 2, 5}

Table 1: Hyper-parameters Tuned for Different Machine Learning Models and Their Values

As showed in Table 1, I tuned multiple hyper-parameters for each ML model, I made sure that each parameter had a wide range in order to find the best parameters for each model. Three random states were used in my pipeline to measure the uncertainty of my evaluation metric due to splitting and due to nondeterministic ML methods(random forest and XGboost).

4 Results

4.1 Baseline and Best Model

As shown in Figure 5, my baseline model has a Macro-Averaged f1 score of 0.032. Random forest and xgboost are the two best performing models. Logistic regression performs worst among all, indicating that there might be a non-linear relationships between my target variable and features. My best model is random forest with 30 for max_depth and 0.25 for max_features, which has a macro f1 score of 0.866 with standard deviation of 0.007. It gives us 120 std above the baseline. All of my models demonstrate strong performance compared to baseline scores, validating the reliability of my approach.

4.2 Confusion Matrix

Something I found particularly interesting while inspecting my model is when I examined the normalized confusion matrix. The diagonal entries represent correctly classified instances, and it's evident that the model performs well for most personality classes. However, there are two distinct parallel lines visible in the matrix. Upon closer inspection, these lines represent cases where the first two letter of the personality type was misclassified. For instance, the model correctly classifies ISTJ 54% of the time. However, when misclassified, ISTJ is predicted as ESTJ 31% of the time and as INTJ 8% of the time. This indicates that the model accurately predicts the last two letters of the personality types but struggles with the first two letters.

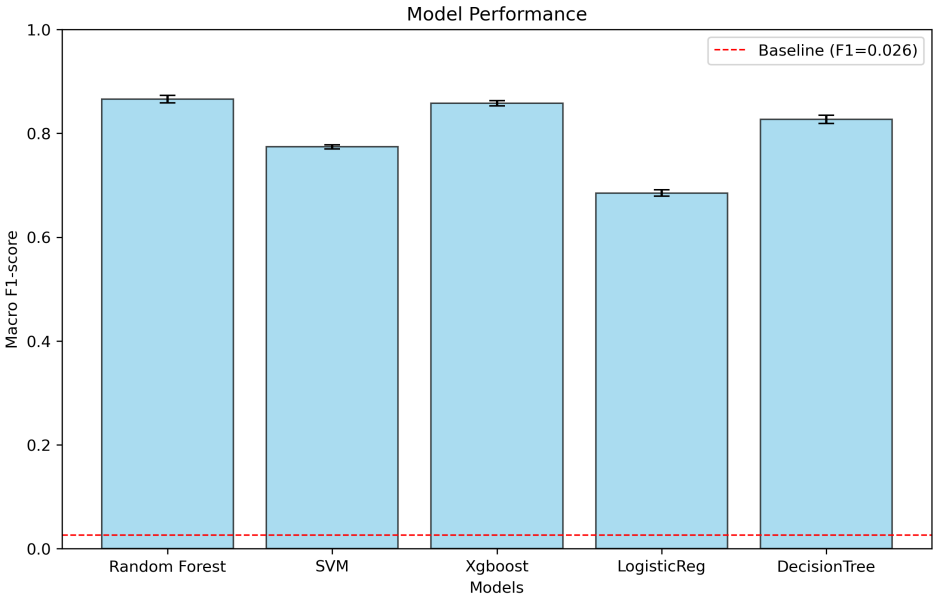

Figure 4: 5 machine learning model performance compared with baseline score

4.3 Three Global Feature Importance

I utilized three ways to measure global feature importance: permutation feature importance, shap summary plot and gini importance. Permutation importance emphasizes the predictive power of features, In my perturbation feature importance figure, the red dash line shows my base line, we can see the model performance drops the most when we shuffle thinking, introversion, sensing and judging score. Thus, based on feature perturbation importance score, we can say that these four scores are the most important features.

Gini importance measures how much a feature contributes to reducing the impurity (Gini index) in the dataset during the training process, given a better feature importance for random forest. Features with higher importance values play a more significant role in the model's decision-making process. We can observe that thinking, introversion, sensing and judging score are the most influential features. Features like one hot interest in technology and others have much less impact on prediction. This results align with perturbation feature importance pretty well.

Another global feature importance metric I used is random forest feature importance, which has very similar shape as gini importance. Agreeing with these four scores are the most important features in my dataset.

In all three plots, the Thinking Score and Judging Score are consistently shown as the two most important features. Depending on the method used to measure feature importance, Sensing Score, Judging Score, and Education alternate as the next three most important features.

4.4 Local Feature Importance

After inspecting the global feature importance, I examine local importance in depth. I investigated how the first point was classified. In Figure 10, we can see that the probability that this point

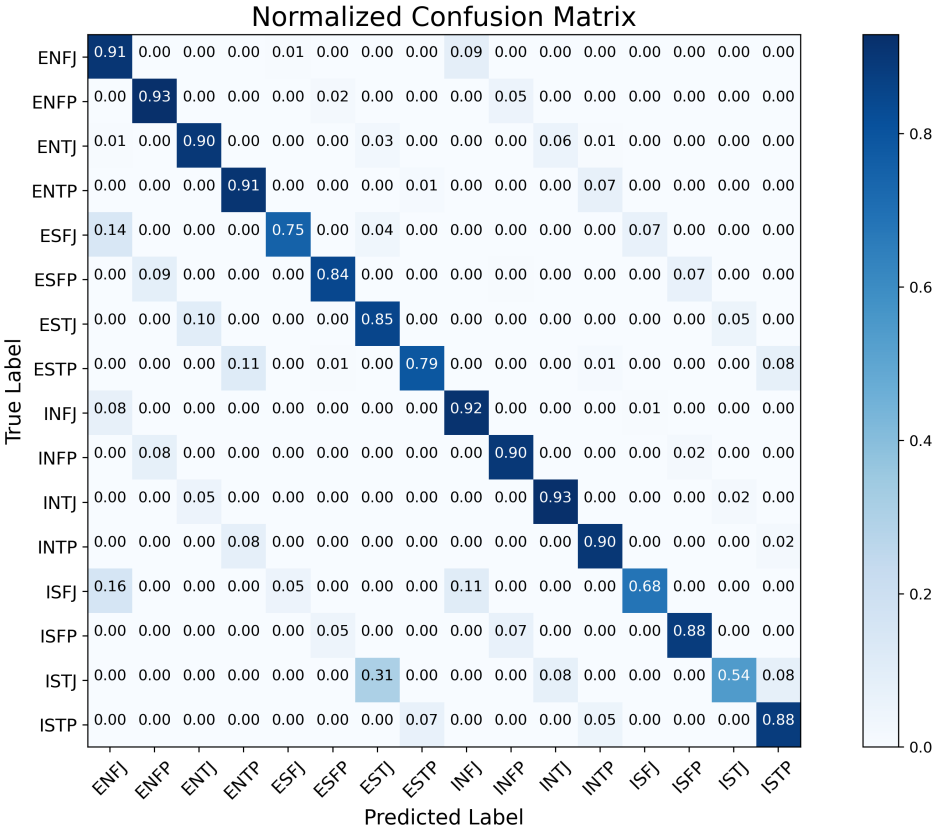

Figure 5: Normalized confusion matrix, where personalities with less data points are less accurately classified

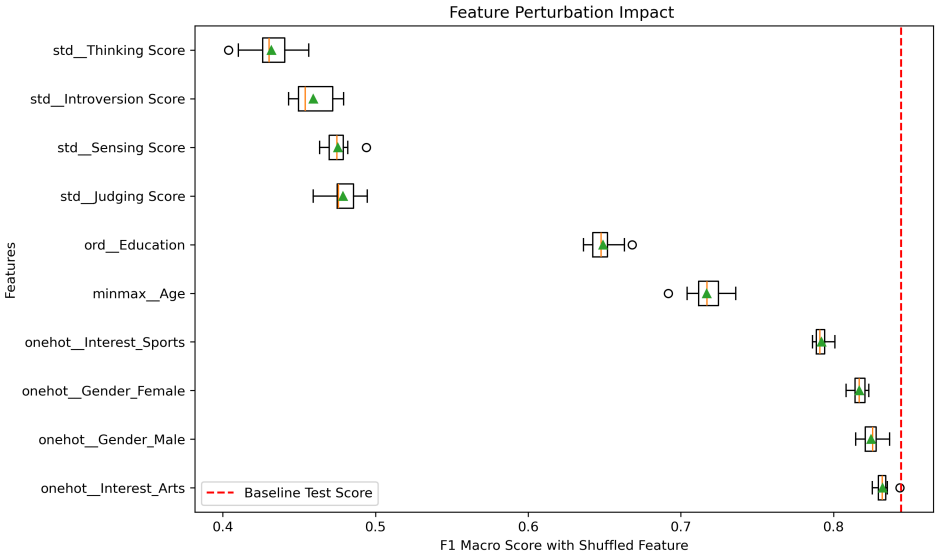

Figure 6: All four scores have the relative large impact on the model performance

belongs to class 6 is 0.01, which is really low. The red features are pushing the prediction higher and have a positive impact on the prediction. The blue features are essentially pushing backwards towards 0. In figure 9, we can see almost all features are pushing the point higher.

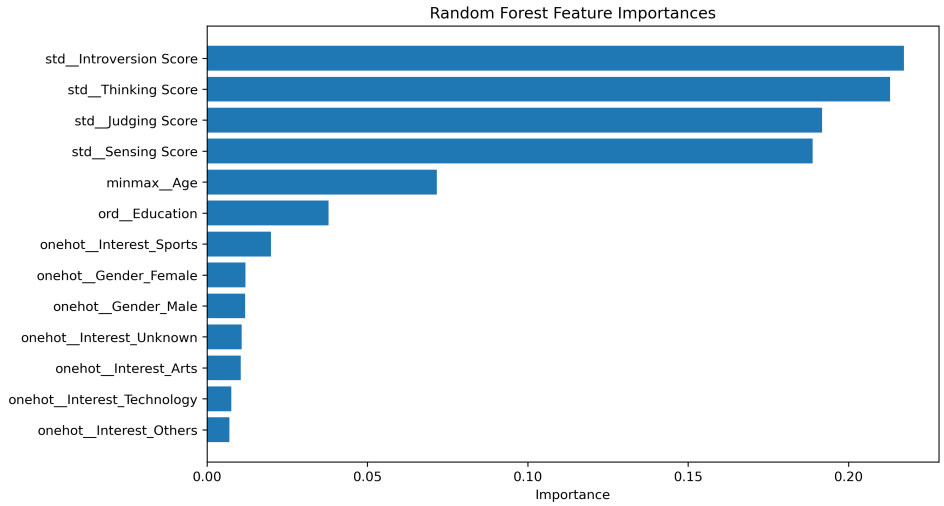

Figure 7: Introversion Score Has the Highest Random Forest Feature Importance

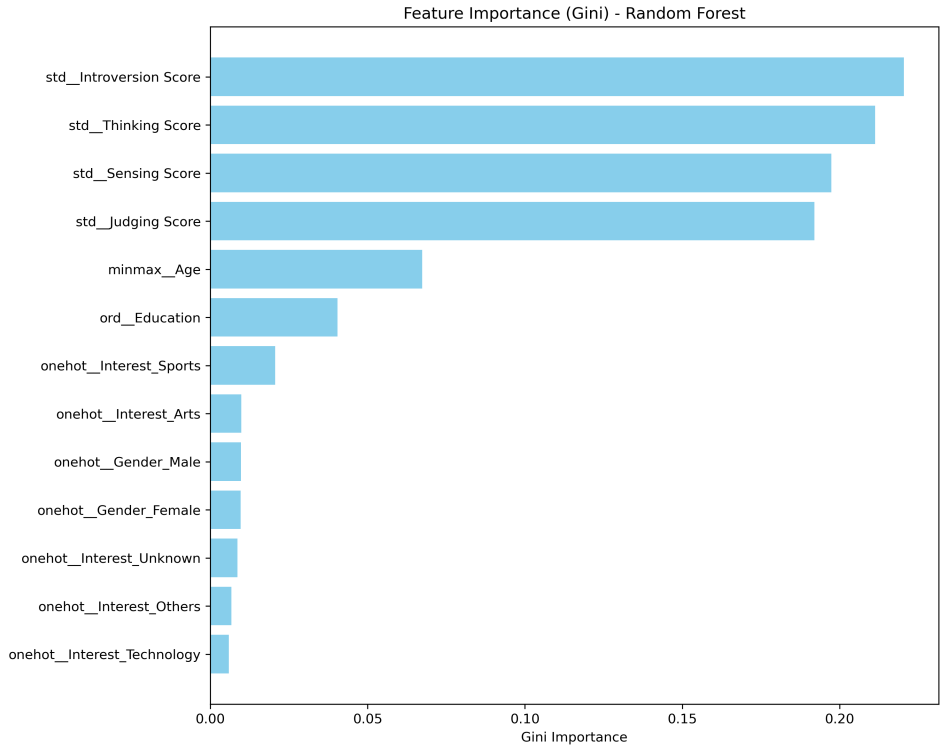

Figure 8: Gini Importance

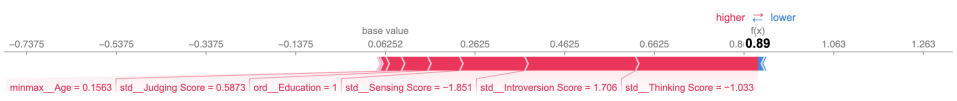

Figure 9: Shape Force Plot for Predicting First Point for Class 2

5 Outlook

Given that the observation about how my model make mistakes on the first two letter more frequently rather than the last two letters. I would like to explore alternative evaluation metrics or adjust the existing metric to include a penalizing term for miss-classifying the first two letter. This

Figure 10: Shape Force Plot for Predicting First Point for Class 6

adjustment would place greater emphasis on penalizing the model when it misclassify the first two letters, as these represent critical aspects of the personality prediction.

There are several weak spots in my modeling approach. Since my dataset is imbalanced in some classes, thus I would try out more reliable sampling techniques instead of SMOTE. Integrating sampling techniques into the modeling pipeline can help balance the dataset and ensure the model learns from all classes equally to improve the robustness of the model. Additionally, my current interpretation methods ignore for potential interactions between features. Exploring combinations of two or three features through permutation could provide deeper insights into how the model utilizes these interactions to make predictions.

To enhance the accuracy and depth of personality predictions, I aim to increase the dimensionality of the dataset by incorporating a broader range of features that provide more comprehensive information about an individual. This includes data on communication styles, social interactions, work environment,and stress responses. By increasing the dimensionality of the dataset in this way, the model can capture richer and more nuanced patterns, potentially leading to more accurate and insightful predictions about an individual's personality.

6 Reference

1. Stealth Technologies. Predict people personality types [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/stealthtechnologies/predict-people-personality-types/data.

2. The Myers & Briggs Foundation. The 16 MBTI personality types. Retrieved from https://www.myersbriggs.org mbti-personality-type/the-16-mbti-personality-types/.

3. GitHub Repository: Huang, J. (2024). Data 1030 project [Repository]. GitHub. Retrieved from https://github.com/huang-0505/data1030-project