Multidimensional Personality Cluster Prediction

AIT 511 / Machine Learning

Assignment 2: Multinomial

Project Overview

Challenge

Human behavior is shaped by a complex interaction of environment, experiences, habits, and personal interests. In this competition, task is to build a machine learning model that predicts an individual's personality cluster based on their behavioral and lifestyle attributes. References [1]

Dataset

Each participant in the dataset is represented by features describing daily routines, activity levels, social engagement, and expressive tendencies. The target variable is a personality cluster label, which represents a group of individuals who share similar behavioral patterns.

Data set folder taken from kaggle includes

1. train.csv: **1913 x 14**
2. test.csv: **479 x 13**

Column/feature Name	**Type**	**Meaning**
participant_id	int 64 (Index column)	Unique ID assigned to each participant
record_code	int 64	Internal reference code (irrelevant to prediction)
age_group	int 64	Age grouping indicator
identity_code	int 64	Encoded personal identity category
cultural_background	int 64	Regional or cultural background grouping
upbringing_influence	int 64	Influence of formative environment
focus_intensity	float 64	Time/effort dedicated toward focused tasks
consistency_score	int 64	Reliability and routine-stability measure
external_guidance_usage	int 64	Use of guidance, mentoring, or support resources
support_environment_score	int 64	Perceived supportive environment level
hobby_engagement_level	int 64	Engagement in leisure or personal interest activities
physical_activity_index	int 64	Physical activity involvement
creative_expression_index	int 64	Participation in artistic or expressive activities
altruism_score	int 64	Tendency toward volunteering or helping behaviors
personality_cluster	object (Target column)	Personality segment label derived externally

Table 1: Dataset Feature Description

Data Processing Steps

Step 1 : Import Libraries and Load Dataset

The initial step in the EDA process involves importing essential libraries such as pandas, numpy, matplotlib, seaborn, and relevant modules from scikit-learn for data preprocessing. The dataset is loaded into a pandas DataFrame for further analysis and manipulation.

'Libraries Used'

```python
import pandas as pd
import numpy as np
import math
```

```python
from scipy.stats import gaussian_kde
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler, RobustScaler
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score, classification_report

from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

import warnings
warnings.filterwarnings("ignore")

```

Step 2: Feature Grouping

Category	Feature Name
Categorical	cultural_background
Numerical	age_group upbringing_influence focus_intensity consistency_score support_environment_score
Binary	identity_code external_guidance_usage hobby_engagement_level physical_activity_index creative_expression_index altruism_score

Table 2: Feature Classification by Category

> **Note**
>
> Here categorical doesn't mean data set is categorical it is assumed as categorical but has numerical values from min 0 to max 3.

Step 3 : Exploratory Data Analysis (EDA)

Various plots were generated to analyse the data distribution, detect outliers, and explore relationships among features. These visualisations provide critical insights that inform subsequent preprocessing and modelling decisions. Key visualisations and their interpretations are detailed below:

Correlation Heat Map

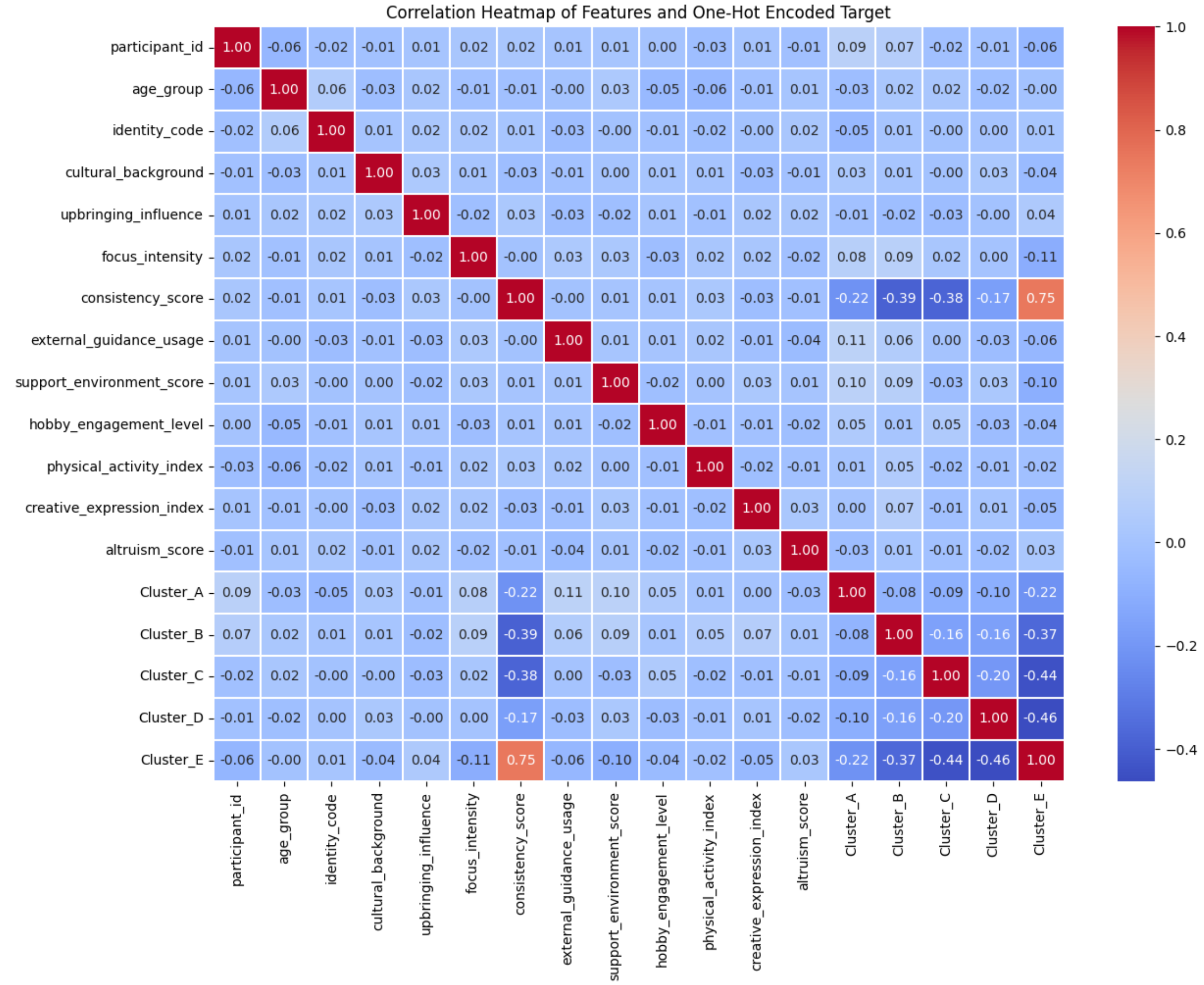

Figure 1: Correlation heat map

The correlation heatmap reveals that most input features exhibit weak linear relationships with the personality clusters, indicating that the classification boundaries are predominantly non-linear in nature. The feature consistency_score shows a strong positive correlation (0.75) with Cluster_E, making it the most influential predictor in the dataset. Strong negative correlations among cluster labels confirm their mutual exclusivity due to correct one-hot encoding. The absence of high inter-feature correlations indicates negligible multicollinearity, ensuring model stability. Several socio-behavioral features such as creative_expression_index, physical_activity_index, and altruism_score show minimal linear contribution, though they may still influence predictions through non-linear models.

Key Takeaways

- Consistency_score is strongly positively correlated with Cluster_E (0.75), making it a key feature for this target group.
- Consistency_score shows strong negative correlation with Clusters B, C, and D, suggesting lower consistency aligns with those clusters .
- Other feature correlations with clusters are generally weak, indicating limited linear relationships for most variables.

Scattered Plots

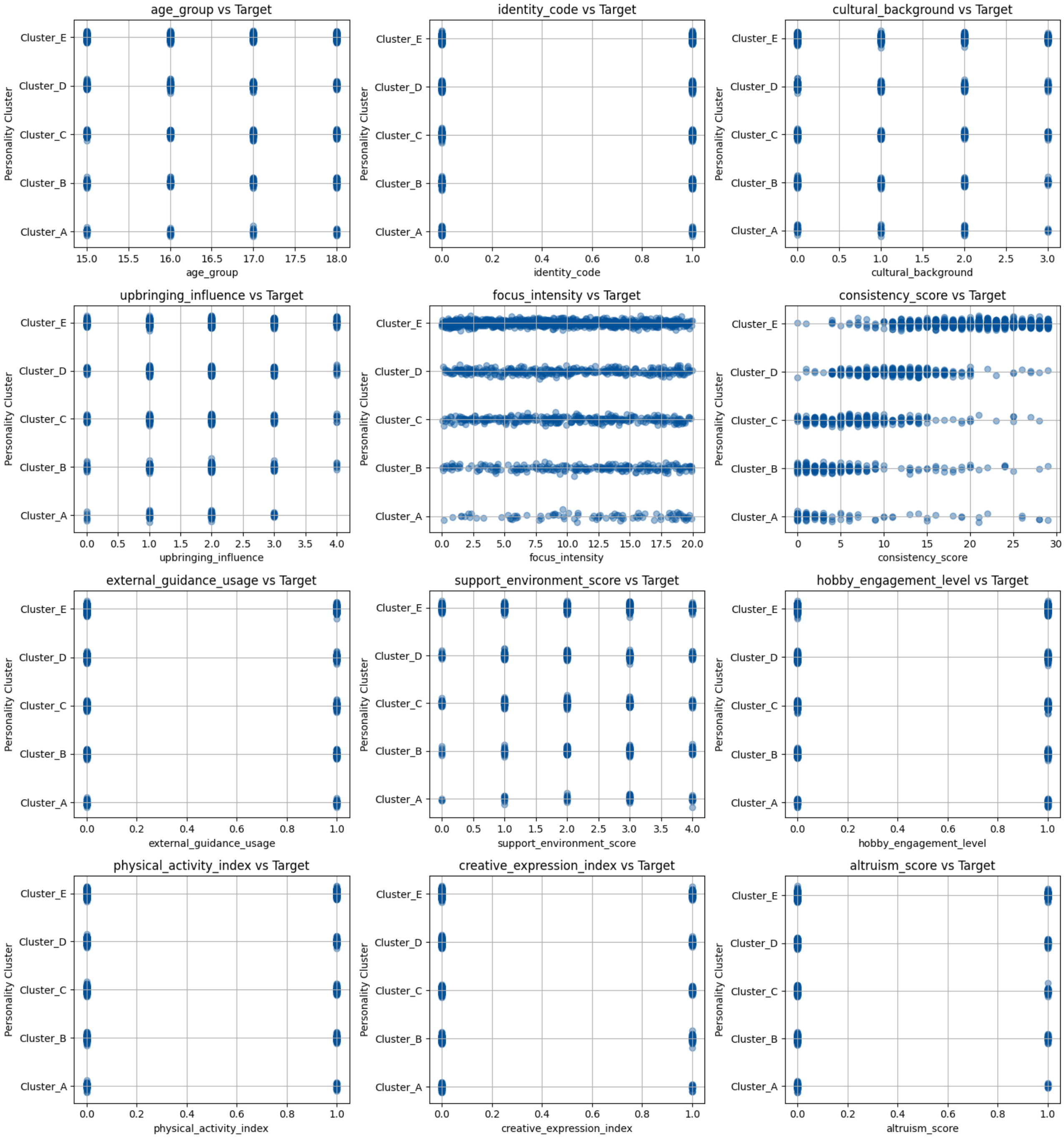

Figure 2: Scattered plot All features

Key Takeaways

- The scatter plots show heavy overlap among all personality clusters, indicating weak linear separability in the feature space.
- Focus intensity and consistency score exhibit a mild positive association, but the relationship is not strong enough to cleanly separate clusters.
- No clearly isolated cluster regions are observed, confirming that the classification task is inherently noisy

and non-linearly separable.
✔ The dense overlap in scatter space explains why even advanced neural network ensembles saturate around 0.58 accuracy.
✔ Overall, the personality clusters appear to be defined by subtle multi-feature interactions rather than simple pairwise relationships.

Histogram

For each numerical feature, histograms and kernel density estimates (KDEs) were plotted to observe the data's spread, skewness, and distribution shape.

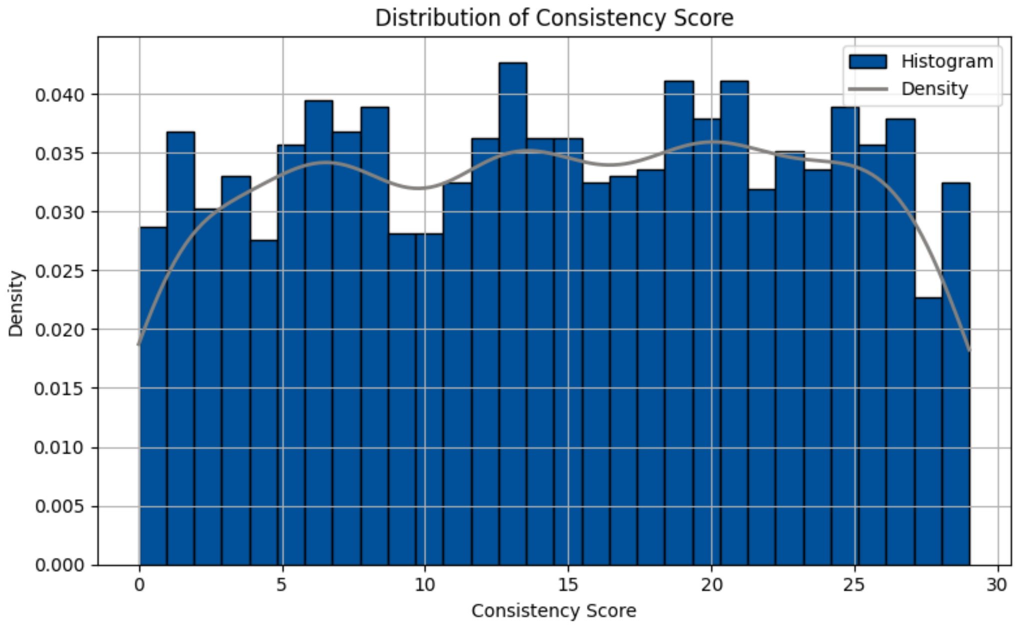

Figure 3: Consistency score distribution and its KDE

Key Takeaways

✔ The consistency score exhibits a right-skewed distribution with a strong concentration at lower values. The smooth density curve confirms the non-Gaussian nature of the feature, indicating potential class overlap in regions of high density.

Boxplots

Boxplots were used to examine each numerical feature for potential outliers, which appear as points beyond the plot's whiskers. See Figure 4.

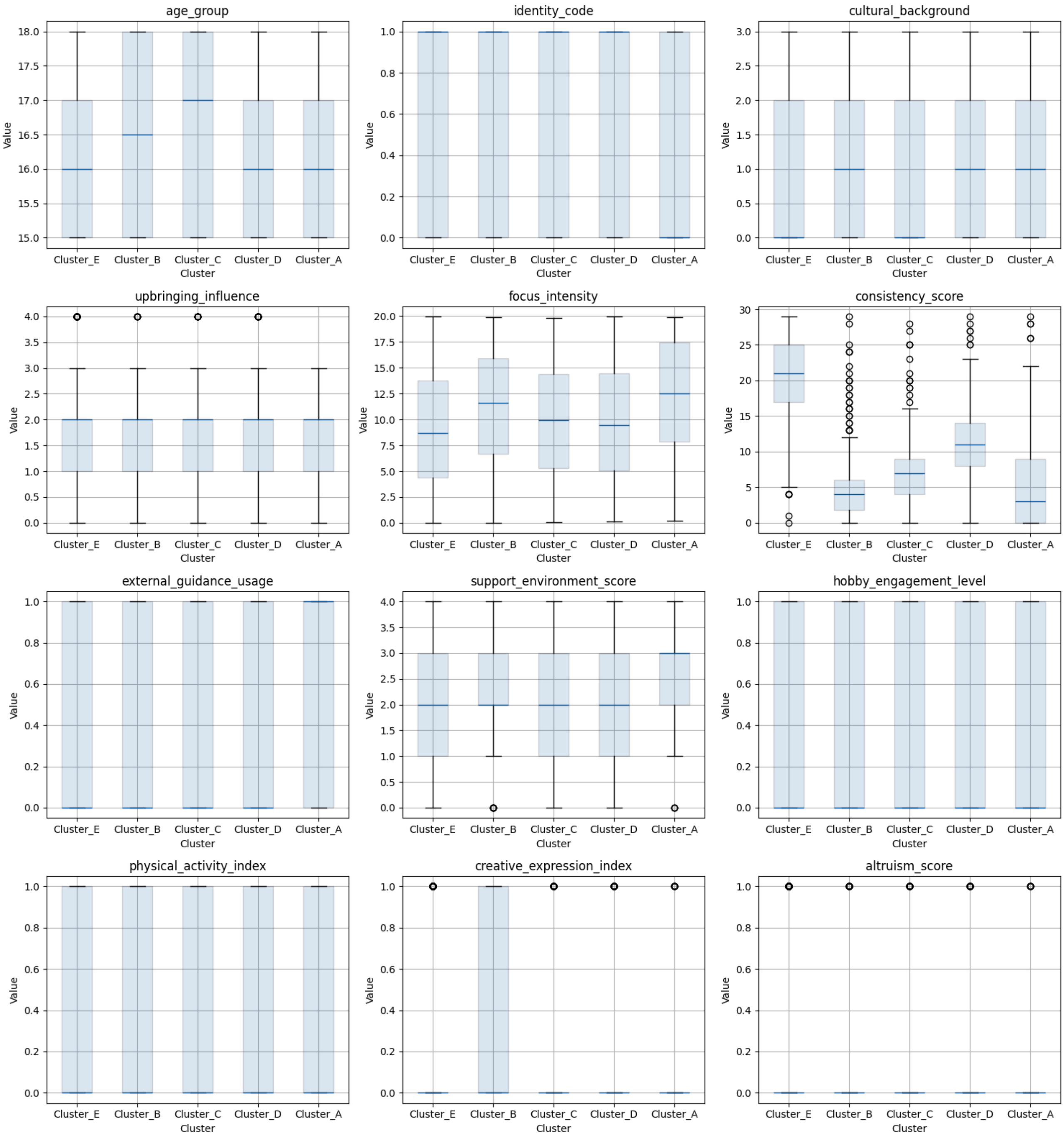

Figure 4: Box Plot for all features

Key Takeaways

Age group, identity code, and cultural background show almost identical distributions across all clusters, indicating very low discriminative power for classification.

- Consistency score is the most informative feature, showing clear separation among clusters, especially distinguishing Cluster D and Cluster E from others.
- Focus intensity shows moderate variation between clusters, suggesting it contributes useful but not strongly decisive information.
- Support environment score shows slight cluster-dependent shifts, but with significant overlap, limiting its standalone predictive strength.
- Upbringing influence exhibits similar medians and spreads across clusters, hence provides minimal separability.
- External guidance usage, hobby engagement level, physical activity index, creative expression index, and altruism score are highly skewed toward a single value, making them weak predictors.
- The large overlap across most features confirms strong class mixing, explaining why model accuracy saturates around 0.58 despite advanced neural ensembles.
- Overall, the personality clusters appear to be driven primarily by behavioral consistency and focus rather than demographic or binary lifestyle indicators.

Step 4 : Data Preprocessing

Data Preprocessing includes:

1. Feature Engineering
2. Handling missing values
3. Scaling of dataset
4. Encoding

Feature Engineering

We applied feature engineering for **focus_intensity** and **consistency_score**.

S.No	Feature Name	Type	Meaning
1	focus_intensity	float64 (Original)	Original raw focus intensity score
2	focus_squared	float64 (Engineered)	Square of focus intensity: $(\text{focus\_intensity})^2$
3	log_focus	float64 (Engineered)	Log-transformed focus: $\log(1 + \text{focus\_intensity})$
4	focus_normalized	float64 (Engineered)	Min–Max scaled focus intensity
5	focus_zscore	float64 (Engineered)	Standardized focus intensity (zero mean, unit variance)
6	focus_consistency_interaction	float64 (Engineered)	Interaction between focus and consistency: $\text{focus\_intensity} \times \text{consistency\_score}$

Table 3: Derived Features from `focus_intensity`

Handling missing values

NO missing values.

Scailing and Encoding of Data set

We have tried :

1. Standard Scaling + One-Hot Encoding
2. Min–Max Scaling + One-Hot Encoding
3. Robust Scaling + One-Hot Encoding
4. Label Encoding Only

S.No	Feature Name	Type	Meaning
1	focus_consistency	float64 (Engineered)	Interaction between focus and consistency computed as focus_intensity $\times$ consistency_score, capturing stable focus behavior
2	support_guidance	float64 (Engineered)	Average of support environment and external guidance usage: $\frac{\text{support\_environment\_score} + \text{external\_guidance\_usage}}{2}$
3	creative_hobby_mean	float64 (Engineered)	Mean of creative expression and hobby engagement: $\frac{\text{creative\_expression\_index} + \text{hobby\_engagement\_level}}{2}$
4	activity_strength	float64 (Engineered)	Overall activity strength computed as the mean of physical activity, hobby engagement, and creative expression
5	stability_mean	float64 (Engineered)	Average of consistency score and support environment score representing behavioral stability
6	guidance_ratio	float64 (Engineered)	Ratio of external guidance usage to support environment: $\frac{\text{external\_guidance\_usage}}{1 + \text{support\_environment\_score}}$

Table 4: Engineered Interaction Feature Description

Step 5 : Training-Validation Split

The dataset is split into training and testing sets, ensuring that model performance can be evaluated on unseen data. A common split ratio is 80% for training and 20% for testing. This step prepares the dataset for model training and evaluation.

'Split Data into Training and Validation'

```
# Split processed data into 80% train and 20% validation
X_train, X_val, y_train, y_val = train_test_split(
    X_processed_df,  # processed features
    y,               # target
    test_size=0.2,   # 20% validation, 80% training
    random_state=42  # for reproducibility
)

# Check shapes
print("X_train:", X_train.shape)
print("X_val:", X_val.shape)
print("y_train:", y_train.shape)
print("y_val:", y_val.shape)
```

Models Used and Hyperparameter tuning

Two supervised learning models were employed for personality cluster classification: Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) Neural Network, both selected due to their strong performance on non-linear, multi-class problems.

Support Vector Machine (SVM)

An SVM with a radial basis function (RBF) kernel was used. Hyperparameter tuning was performed over the following ranges:

- Regularization parameter: $C \in \{0.1, 1, 10, 50, 100\}$
- Kernel coefficient: $\gamma \in \{0.01, 0.05, 0.1, \text{scale}\}$

To address class imbalance, class-weighted SVM was applied. Model selection was based on the macro F1-score evaluated on the validation set.

Multi-Layer Perceptron (MLP)

A feed-forward neural network with ReLU activation was trained. The following hyperparameters were tuned:

- Hidden layer sizes: (64), (128), (256), $(128, 64)$, $(256, 128)$, $(256, 128, 64)$
- Learning rates: $\{0.001, 0.0005\}$
- L2 regularization parameter: $\alpha \in \{10^{-5}, 10^{-4}, 10^{-3}\}$

Early stopping was employed to prevent overfitting. Each configuration was evaluated using the macro F1-score.

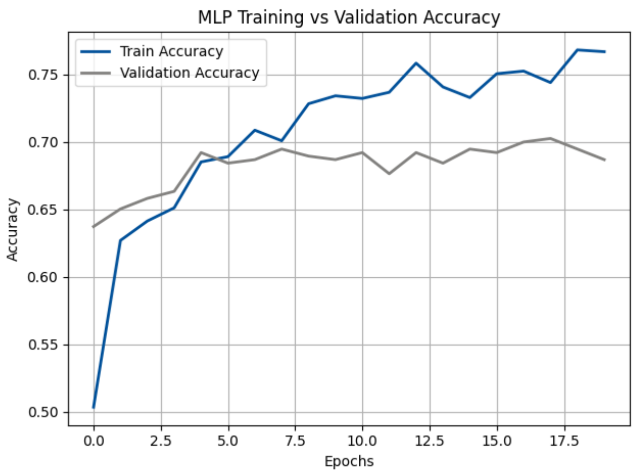

Figure 5: Training and validation accuracy curves across epochs for the MLP model corresponding to the submission `mlp_submission_v1`. The plot shows steady learning with stable validation performance, indicating controlled overfitting and good generalization.

→ As the score was so-so, we wanted to increase the score. so we tried looking at the target column cluster count, then we observed unbalanced distribution schown below so we tried to normalise it. But our score got even worse. So we dropped it.

→ Next we tried neural network k-fold which gave a decent score.

Ensemble

We also tried ensembling. But it couldn't beat the nn_kfold we performed.

S.No	Personality Cluster	Proportion
1	Cluster_E	0.5091
2	Cluster_D	0.1715
3	Cluster_C	0.1600
4	Cluster_B	0.1150
5	Cluster_A	0.0444

Table 5: Class Distribution of Personality Clusters

Model Selection

All trained SVM and MLP configurations were ranked based on their validation macro F1-scores. The top five performing models were selected for final prediction and submission generation.

Summary

S.No	Submission File	Score	Remarks
1	**mlp_submission_v10**	**0.627**	**Base features with tuned MLP.**
2	**mlp_submission_v11**	**0.627**	**Alternate learning rate configuration.**
3	mlp_submission_v12	0.601	Standard scaling with base features.
4	mlp_submission_v13	0.612	Increased depth with regularization.
5	mlp_submission_v14	0.601	Reduced-layer MLP architecture.
6	mlp_submission_v15	0.496	Interaction features without retuning.
7	mlp_submission_v16	0.507	Engineered features with early stopping.
8	mlp_submission_v17	0.588	Feature engineering with robust scaling.
9	mlp_submission_v18	0.588	Deeper network with engineered features.
10	mlp_submission_v19	0.557	Compact MLP on engineered inputs.
11	logistic_submission	0.453	Linear baseline with standardized features.
12	svm_submission_v1	0.538	RBF SVM with scaled inputs.
13	svm_pcs_submission	0.518	PCA-reduced features with SVM.
14	svm_mlp_ensemble	0.544	Soft-voting ensemble of classifiers.
15	svm_mlp_prob_ensemble	0.511	Probability-averaged model ensemble.
16	nn_kfold_ensembel_v2	0.577	K-fold neural network aggregation.
17	nn_kfold_ensembel_v3	0.562	K-fold ensemble with tuned depth.
18	nn_multirun_ensemble	0.549	Multi-run averaged neural predictions.
19	stacking_submission	0.512	Meta-learner based stacked model.
20	naive_bayes	0.384	Gaussian Naive Bayes classifier.

Table 6: Summary of Submission Files and Model Variants

We found out that the mlp (256, 128, 64) with label coding only i.e. mlp_submission_v10 gave us the best score for the leaderboard part.

Team Members

Project Katakam - (Team Name)

S.No.	Name (Roll No.)	Role
1	Mohit Jagini (IMT2023528)	Member
2	Katakam Shashidhar Sai (IMT2023567)	Team Leader
3	Hardhik Dhavala (IMT2023579)	Member

> **Note**
>
> Click here to email the team leader with team members in CC

References

[1] Kaggle, "Multidimensional personality cluster prediction challenge." https://www.kaggle.com/competitions/multidimensional-personality-cluster-prediction/overview, 2025. Kaggle Competition.

[2] P. Katakam, "Multidimensional personality cluster prediction challenge." https://github.com/DHardhik/ML-Challenge, 2025. Accessed on November 27, 2025.