Human Personality Classification Using Machine Learning and Deep Learning

By Kowsik.S

Table of Contents

Abstract

This Capstone project focuses on classifying human personality types (Introvert vs. Extrovert) using machine learning (ML) and deep learning (DL) techniques. The dataset, comprising social behavior attributes, was used to train Logistic Regression, Random Forest, and Artificial Neural Network (ANN) models. Extensive preprocessing addressed missing values and class imbalance, ensuring robust model performance. The Random Forest model achieved the highest accuracy at 95%, followed by Logistic Regression at 92%, and the ANN at 90%. This report details the preprocessing steps, missing value treatment, ANN implementation, and model evaluations, highlighting the effectiveness of ML and DL in personality classification.

Introduction

Personality classification is a valuable tool in psychology, human resources, and social sciences. This project aims to predict whether an individual is an introvert or extrovert based on social behavior attributes using ML and DL models. The dataset includes features like time spent alone, stage fear, and social event attendance. Key challenges included handling missing data, addressing class imbalance, and implementing an ANN for comparison with traditional ML models. This report provides a detailed analysis of preprocessing, missing value treatment, ANN implementation, and results.

Data Description

The dataset contains 2,900 entries with eight columns: seven features and one target variable (Personality: Introvert or Extrovert). The features are:

- **Time_spent_Alone**: Hours spent alone (numeric, 0–10).
- **Stage_fear**: Presence of stage fear (categorical: Yes/No).
- **Social_event_attendance**: Frequency of attending social events (numeric, 0–10).
- **Going_outside**: Frequency of going outside (numeric, 0–10).
- **Drained_after_socializing**: Feeling drained after socializing (categorical: Yes/No).
- **Friends_circle_size**: Number of close friends (numeric, 0–15).
- **Post_frequency**: Frequency of social media posts (numeric, 0–10).

Missing values were present in all features except the target, with counts ranging from 52 to 77 per column.

Methodology

Data Preprocessing

- **Feature Categorization**: Features were divided into numeric (Time_spent_Alone, Social_event_attendance, Going_outside, Friends_circle_size, Post_frequency) and categorical (Stage_fear, Drained_after_socializing) for targeted preprocessing.
- **Encoding Categorical Variables**: LabelEncoder was used to convert categorical variables to numeric values. For Stage_fear and Drained_after_socializing, ‘Yes’ was encoded as 1 and ‘No’ as 0. The target variable Personality was encoded as Introvert=1 and Extrovert=0.
- **Feature Scaling**: Numeric features were scaled using StandardScaler to standardize their range, ensuring compatibility with gradient-based models like ANN and Logistic Regression. Scaling was applied after the train-test split to prevent data leakage.
- **Train-Test Split**: The dataset was split into 80% training (2,320 samples) and 20% testing (580 samples) sets using train_test_split with a random state for reproducibility.
- **Class Imbalance Handling**: The dataset exhibited class imbalance, with introverts and extroverts unevenly distributed. The Synthetic Minority Oversampling Technique

(SMOTE) was applied to the training set to balance classes by generating synthetic samples for the minority class, ensuring equal representation.

Missing Value Treatment

- **Numeric Features**: Missing values in Time_spent_Alone (63 missing), Social_event_attendance (62 missing), Going_outside (66 missing), Friends_circle_size (77 missing), and Post_frequency (65 missing) were imputed using the median. The median was chosen due to the presence of skewness in these features, as observed in distribution plots generated with seaborn.distplot. The SimpleImputer with strategy='median' was applied.
- **Categorical Features**: Missing values in Stage_fear (73 missing) and Drained_after_socializing (52 missing) were imputed using the mode (most frequent value) with SimpleImputer (strategy='most_frequent'). This approach ensured consistency with the categorical nature of these variables.
- **Verification**: Post-imputation, the dataset was checked for missing values using data.isnull().sum(), confirming no missing data remained.

Exploratory Data Analysis

- Time_spent_Alone and Friends_circle_size showed moderate right skewness, justifying median imputation.
- Social_event_attendance and Post_frequency had near-normal distributions, supporting their use in predictive modeling.

These insights guided the preprocessing strategy, ensuring robust feature preparation.

Sample Model Selection

- **Logistic Regression**: A baseline linear model for binary classification.
- **Random Forest Classifier**: An ensemble model to capture non-linear relationships.
- **Artificial Neural Network (ANN)**: A deep learning model to explore complex patterns.

ANN Implementation

The ANN was implemented using TensorFlow/Keras with the following architecture:

- **Input Layer**: 7 neurons corresponding to the 7 features.
- **Hidden Layers**: Two dense layers with 16 and 8 neurons, respectively, using ReLU activation to capture non-linearities.
- **Output Layer**: 1 neuron with sigmoid activation for binary classification (Introvert vs. Extrovert).
- **Compilation**: The model was compiled with the Adam optimizer, binary cross-entropy loss, and accuracy as the metric.
- **Training**: The model was trained for 50 epochs with a batch size of 32, using the SMOTE-balanced training data. A validation split of 20% was used to monitor performance.

The ANN was scaled using StandardScaler to ensure optimal gradient descent performance.

Evaluation Metrics

Models were evaluated using:

- **Accuracy**: Proportion of correct predictions.
- **Precision**: Ratio of true positives to predicted positives.
- **Recall**: Ratio of true positives to actual positives.
- **F1-Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: Visualized using seaborn.heatmap to assess prediction errors.

Cross-validation was performed using cross_val_score to ensure robustness.

Results

Logistic Regression

- **Accuracy**: 92%.
- **Precision**: 0.91 (Introvert), 0.93 (Extrovert).
- **Recall**: 0.93 (Introvert), 0.91 (Extrovert).
- **F1-Score**: 0.92 (average).

The confusion matrix showed balanced performance with few misclassifications.

Random Forest Classifier

- **Accuracy**: 95%.
- **Precision**: 0.94 (Introvert), 0.96 (Extrovert).
- **Recall**: 0.96 (Introvert), 0.94 (Extrovert).
- **F1-Score**: 0.95 (average).

Random Forest outperformed Logistic Regression, with fewer errors in the confusion matrix.

Artificial Neural Network

- **Accuracy**: 90%.
- **Precision**: 0.89 (Introvert), 0.91 (Extrovert).
- **Recall**: 0.90 (Introvert), 0.90 (Extrovert).
- **F1-Score**: 0.90 (average).

The ANN performed well but was slightly less accurate than Random Forest, likely due to the dataset's size and feature complexity.

Sample Predictions

- **Sample 1** (Time_spent_Alone=9, Stage_fear=Yes, Social_event_attendance=0, Going_outside=0, Drained_after_socializing=Yes, Friends_circle_size=0, Post_frequency=3): Predicted as **Introvert** by Logistic Regression and ANN.
- **Sample 2** (Time_spent_Alone=4, Stage_fear=No, Social_event_attendance=4, Going_outside=6, Drained_after_socializing=No, Friends_circle_size=13, Post_frequency=5): Predicted as **Extrovert** by Random Forest and ANN.

Discussion

Random Forest achieved the highest accuracy (95%) due to its ability to handle non-linear relationships and feature interactions. Logistic Regression (92%) served as a robust baseline, while the ANN (90%) showed promise but was limited by the dataset's size, which may not fully leverage deep learning's potential. SMOTE effectively mitigated class imbalance, and median/mode imputation preserved data integrity. Limitations include potential overfitting in Random Forest and the ANN's sensitivity to hyperparameter tuning.

Future work could involve larger datasets, additional features, or advanced DL architectures like LSTMs.

Conclusion

This project demonstrated the effectiveness of ML and DL in classifying personality types based on social behavior. Random Forest was the most accurate model, followed by Logistic Regression and ANN. The preprocessing pipeline, including missing value treatment and SMOTE, was critical to success. These findings suggest practical applications in psychological profiling and personalized recommendations, with potential for further exploration using advanced DL techniques.

References

1. Scikit-learn: Machine Learning in Python. https://scikit-learn.org
2. TensorFlow Documentation. https://www.tensorflow.org
3. Pandas Documentation. https://pandas.pydata.org
4. Seaborn: Statistical Data Visualization. https://seaborn.pydata.org