

Personality Prediction Project

This project aims to predict personality traits (Extrovert/Introvert) based on various social and personal metrics using machine learning models. The process involves data loading, preprocessing, training and evaluating multiple classification models, and finally selecting and saving the best performing model.

Table of Contents

- [Dataset](#)
- [Methodology](#)
- [1. Data Loading and Exploration](#)
- [2. Data Preprocessing](#)
- [3. Model Training and Evaluation](#)
- [4. Model Selection and Saving](#)
- [Dependencies](#)
- [Usage](#)

Dataset

The dataset used in this project contains the following features:

- `Time_spent_Alone` : Time spent alone (float)
- `Stage_fear` : Presence of stage fear (object - categorical: Yes/No)
- `Social_event_attendance` : Frequency of social event attendance (float)
- `Going_outside` : Frequency of going outside (float)
- `Drained_after_socializing` : Feeling drained after socializing (object - categorical: Yes/No)
- `Friends_circle_size` : Size of friends circle (float)
- `Post_frequency` : Frequency of social media posts (float)

- **Personality** : Target variable (object - categorical: Extrovert/Introvert)

Methodology

1. Data Loading and Exploration

- The `personality_dataset.csv` file is loaded into a pandas DataFrame.
- Basic information about the dataset, including data types, non-null counts, and descriptive statistics, is displayed.
- Missing values in each column are checked.

2. Data Preprocessing

This step prepares the raw data for machine learning by handling missing values, encoding categorical features, and transforming the target variable.

- **Target Separation & Feature Type Identification:** The `Personality` column is separated as the target variable (`y`), and the remaining columns as features (`X`). Features are then identified as either categorical or numerical.
- **Preprocessing Pipelines:** Separate pipelines are created for numerical and categorical features:
- **Numerical Pipeline:** Missing values are imputed using the mean strategy.
- **Categorical Pipeline:** Missing values are imputed using the most frequent strategy, followed by One-Hot Encoding.
- **Combine & Apply Preprocessing:** A `ColumnTransformer` is used to combine these pipelines and apply the transformations to the feature set `X`.
- **Encoding Target Variable:** The `Personality` target variable (Extrovert/Introvert) is encoded into numerical form (0/1) using `LabelEncoder`.

3. Model Training and Evaluation

- **Dataset Splitting:** The preprocessed data is split into training and testing sets (80% training, 20% testing) using `train_test_split` with stratification to maintain class distribution.
- **Model Definition and Hyperparameter Grids:** The following classification models are defined along with their hyperparameter grids for `GridSearchCV`:

- Random Forest Classifier
- Gradient Boosting Classifier
- Support Vector Machine (SVM) Classifier
- Logistic Regression
- **Training, Tuning, and Evaluation:** Each model is trained and tuned using `GridSearchCV` with 5-fold cross-validation. Performance metrics such as accuracy, precision, recall, and F1-score are calculated and stored for comparison.

4. Model Selection and Saving

- **Best Model Selection:** The model with the highest cross-validation accuracy is identified as the best model.
- **Model Saving:** The best performing model is saved using `joblib` for future use.

Dependencies

The project requires the following Python libraries:

- `pandas`
- `numpy`
- `scikit-learn` (`sklearn`)
- `pickle`
- `joblib`

To install these dependencies, you can use pip:

```
pip install pandas numpy scikit-learn joblib
```

Usage

To run this notebook and reproduce the results:

1. Ensure you have all the dependencies installed.
2. Place your `personality_dataset.csv` file in a `data` directory relative to the notebook.

3. Open the `pasindu_final_notebook.ipynb` file in a Jupyter environment (Jupyter Notebook, JupyterLab, VS Code with Python extension, etc.).
4. Run all cells in the notebook sequentially.

The notebook will perform data loading, preprocessing, model training, evaluation, and save the best model to a file named `best_model.pkl`.

API Usage (Hypothetical)

While this notebook focuses on model training, the trained model (`best_model.pkl`) could be deployed as a web service to make predictions via an API. Below is a hypothetical example of how you might interact with such an API using a POST request.

Endpoint

`POST /predict_personality`

Sample Request Body

To get a personality prediction, send a JSON object containing the features as shown below. Ensure the data types and column names match the training data.

```
{
  "Time_spent_Alone": 5,
  "Stage_fear": "No",
  "Social_event_attendance": 3,
  "Going_outside": 7,
  "Drained_after_socializing": "No",
  "Friends_circle_size": 7.0,
  "Post_frequency": 2.0
}
```

Expected Response

The API would return a JSON object containing the predicted personality type (Extrovert or Introvert).


```
{
  "prediction": "Introvert"
}
```

Streamlit Application

A user-friendly web application built with Streamlit is available to interact with the trained personality prediction model. You can access it via the following link:


[Personality Predictor Streamlit App](#)


Application Screenshot

 **Intr overt vs. Extrovert Personality Predictor**

Discover your dominant personality trait based on your lifestyle preferences


Please provide the following details:

 **Personal Habits**


 Time spent alone (hours/day)

5

011


 Do you have stage fear?

Yes

 Social event attendance (per month)


5

010


 Frequency of going outside (days/week)

4

07


 Feel drained after socializing?

Yes

 Number of close friends

7

015

 Social media post frequency (per week)

5

010

Predict My Personality