Graphic Era deemed to be University
DEHRADUN

PROJECT AND TEAM INFORMATION

Project Title

AI-Powered Personality Prediction

Student/Team Information

Team Name:	DAA-IV-T102
Team member 1 (Team Lead) (Last Name, name: student ID: email, picture):	Tripathi, Shorya – 230213752 Shoryatripathi0606@gmail.com 
Team member 2 (Last Name, name: student ID: email, picture):	Tushar – 23021717 23021717@geu.ac.in 

Team member 3 (Last Name, name: student ID: email, picture):	Kumari, Sagun – 23012356 shmehta109@gmail.com 

PROJECT PROGRESS DESCRIPTION (35 pts)

Project Abstract (2 pts)

This project focuses on creating an intelligent system that predicts personality traits using AI-based text analysis. Based on the Big Five personality model (OCEAN), the system processes user-generated content—such as tweets or diary entries—using Natural Language Processing (NLP). It extracts statistical and semantic features from the text using tools like TF-IDF, Word2Vec, and BERT, then uses classification models such as Logistic Regression, SVM, or Random Forest to infer traits like Openness and Neuroticism. The goal is to develop an accessible, automatic, and scalable method of personality assessment, with applications in mental health, hiring, content personalization, and more.

Updated Project Approach and Architecture (2 pts)

The system is built using Python and Jupyter Notebook, utilizing libraries such as Scikit-learn, NLTK, and TensorFlow. It follows a modular pipeline:

- **Text Input Layer:** Accepts social media-like text (Twitter/Reddit format).
- **Preprocessing Layer:** Tokenization, stopword removal, punctuation stripping.
- **Feature Extraction:** Statistical features (sentence length, lexical diversity) and vector embeddings (TF-IDF, Word2Vec, BERT).
- **Statistical Analysis Module:** Correlation and chi-square analysis.
- **Model Training Module:** Trains Logistic Regression, SVM, or Random Forest classifiers.
- **Output Layer:** Visualizes trait predictions with graphs and scores.

Tasks Completed (7 pts)

Task Completed	Team Member
• Designed the overall architecture of the system. • Set up and validated TF-IDF and Word2Vec pipelines. • Managed GitHub and coordinated task distribution.	Shorya Tripathi
• Handled statistical feature extraction (e.g., lexical diversity, average word length). • Built and trained initial classification models using scikit-learn. • Validated model performance (precision, recall).	Tushar
• Preprocessed and cleaned text datasets. • Designed trait distribution visualizations using matplotlib and seaborn. • Managed documentation and prepared visual reports.	Sagun Kumari

Challenges/Roadblocks (7 pts)

- **Dataset Quality**

Social media text is often noisy and short, requiring advanced preprocessing and filtering.

- **Embedding Comparisons**

Choosing between traditional TF-IDF and contextual BERT embeddings posed performance trade-offs.

- **Multilabel Classification**

Predicting all five traits independently required separate modeling strategies.

- **Interpretability**

Making personality outputs human-readable and intuitive through visual graphs was non-trivial.

- **Model Generalization**

Ensuring the model didn't overfit on small or synthetic datasets was a constant concern.

Tasks Pending (7 pts)

Task Pending	Team Member (to complete the task)
• Implement multi-label learning to improve trait interdependence modeling.	Shorya Tripathi
• Integrate BERT embeddings and fine-tune classification layers using TensorFlow/Keras.	Tushar
• Finalize front-end reporting layout and polish output visualization with trait comparisons. .	Sagun Kumari

Project Outcome/Deliverables (2 pts)

The AI-powered personality prediction system is approximately **70% complete** and demonstrates significant functional progress. The project combines machine learning, natural language processing, and psychological theory to deliver a modern solution for digital personality analysis.

✅ **Major Deliverables (Developed or In Progress):**

- **AI-Driven Personality Classifier**
  A multi-model system that predicts Big Five traits based on user-generated text input using both traditional (TF-IDF, Word2Vec) and advanced (BERT) embeddings.
- **NLP Preprocessing Pipeline**
  Fully implemented module that tokenizes, cleans, and processes input text using NLTK and custom filters.
- **Statistical Feature Analyzer**
  Extracts features such as lexical diversity, word/sentence length, frequency of pronouns, emotional words, etc., to enrich classification.
- **Model Evaluation Framework**
  Generates precision, recall, F1-score, and confusion matrices for all models, allowing performance comparison between TF-IDF, Word2Vec, and BERT-based systems.
- **Interactive Visual Output**
  Produces bar plots of trait scores (Openness to Neuroticism), feature importance rankings, and trait distribution comparisons using matplotlib/seaborn.
- **Modular Architecture**
  Codebase is separated into preprocessing, feature extraction, modeling, and visualization modules to ensure scalability and readability.
- **Partial Integration of Deep Learning**
  Transformer-based models (BERT) are under integration for context-aware personality prediction, adding semantic depth to the feature space.
- **Real-World Dataset Compatibility**
  Compatible with the myPersonality dataset format and supports synthetic data augmentation for research purposes.
- **Jupyter Notebook Demonstrations**
  Functional notebooks exist for each module (preprocessing, vectorization, modeling, and visualization) for educational and reproducibility purposes.
- **Documented Pipeline**
  The entire methodology—including preprocessing steps, model logic, evaluation metrics, and dataset structure—is being documented for final submission.

🔄 **Pending/Upcoming Deliverables (Final 30%)**

- Full integration of BERT-based model and its performance tuning.
- Final GUI or web interface to allow user input and display personality graphs dynamically.
- Consolidated report (PDF/HTML) with all performance metrics and trait outputs.
- Comparative analysis with existing tools or surveys for validation.

Progress Overview (2 pts)

Tasks on Schedule:

- Text preprocessing
- Statistical feature extraction
- Classification model setup

Ahead of Schedule:

- Vectorization pipeline using TF-IDF and Word2Vec
- Initial model training and visualization efforts

Behind Schedule:

- BERT integration and optimization
- Multi-trait prediction model
- Final user interface and report generation

Testing and Validation Status (2 pts)

Test Type	Status (Pass/Fail)	Notes
Functionality Test	✅ Pass	All current modules work as expected
Vectorization Test	✅ Pass	TF-IDF, Word2Vec integrated successfully
Model Accuracy Test	✅ Pass	Accuracy > 75% on synthetic and small real dataset
Visualization Output Test	🟡 Partial	Graphical output generation is still under improvement
BERT Embedding Integration	✅ Pass	Requires large dataset and Compute resources

Deliverables Progress (2 pts)

Deliverable	Status	Remarks
Text Preprocessing Module	✅ Completed	Tokenization, cleaning, and normalization implemented using NLTK.
Statistical Feature Extraction	✅ Completed	Lexical diversity, sentence/word length, and frequency metrics extracted.
Vectorization (TF-IDF & Word2Vec)	✅ Completed	Successfully integrated and tested with sample data.
Initial Model Training (Logistic Regression, SVM)	✅ Completed	Achieved 75–80% accuracy on validation data.
Trait Visualization (matplotlib/seaborn)	✅ Completed	Bar plots for Big Five traits and feature importance are functional.
Model Evaluation Module	🔄 In Progress	F1 scores, confusion matrices implemented; ROC curves under development.
Deep Learning (BERT) Integration	🔄 In Progress	Pretrained BERT tokenizer loaded; classifier head being fine-tuned.
Dashboard/Report Generation	🔄 In Progress	Output report formatting and layout under design.
Interactive Input UI / Web Form	❌ Pending	Planned for final milestone; may use Gradio or Flask.
Final Report Documentation	❌ Pending	Structure ready; compilation and formatting to be done post-testing.
System Architecture Diagram	✅ Completed	Block diagram finalized to show workflow from input to output.
Jupyter Notebooks for Demonstration	✅ Completed	Separate notebooks for preprocessing, training, and visualization modules.