Detecting AI-Generated Text — Proposal

Capstone: The Art of Approximation
Ojas Chaturvedi, Ritwik Jayaraman, Saianshul Vishnubhaktula, Zaheen Jamil

Language

Python, a simple and popular language for machine learning and data science due to its extensive libraries and frameworks

Objective

To develop a custom machine learning model which would be able to detect AI-generated text

Implementation

Overview of Steps:

1. Data Exploration
2. Data Preprocessing
3. Feature Engineering
4. Model Development
5. Training and Testing
6. Evaluation
7. Optimization
8. Documentation

Potential Libraries:

- Pandas: For data manipulation and analysis
- NumPy: For numerical computing and working with arrays
- Matplotlib: For data visualization
- Scikit-learn: For data mining and analysis
- TensorFlow: For deep learning and complex neural network modeling
- PyTorch: For natural language processing
- NLTK/spaCy: For human language data with symptom inputs
- Flask/Django: For backend web development
- SQLAlchemy: For SQL databases and Object-Relational Mapping

Manual Work:

- Compiling datasets
- Building custom model
- Training and testing model
- Creating website/app
- Documentation of all steps

Jobs

- Machine Learning Developers
  - Develops the machine learning model
  - Trains & tests the model
  - Makes the model usable in the website/app
- Data Analyst
  - Data-exploration, preprocessing, and feature engineering
  - Will still contribute as a Machine Learning Developer
- GUI Developer
  - Makes the website/app and all of its functionality (UI)
  - Will still contribute as a Machine Learning Developer