LLM HACKATHON SOLUTION

Manoj Kumar

DATA SCIENTIST

To solve this hackathon challenge, we'll need to build a model that can predict whether an essay was written by a human or generated by a language model. We'll follow these steps:

1. **Data Exploration:** Understand the structure of the provided data files.
2. **Data Preprocessing:** Prepare the data for modeling, including combining the prompts with the essays.
3. **Feature Engineering:** Extract features that could be helpful for distinguishing between human-written and AI-generated essays.
4. **Model Building:** Create a machine learning model to make the predictions.
5. **Model Evaluation:** Evaluate the model using the roc_auc metric.
6. **Submission Preparation:** Prepare the submission file according to the guidelines.

The dataset contains the following columns:

id: This seems to be a unique identifier for each row.

prompt_id: This could be an identifier for different prompts or categories.

text: This column contains textual data, likely responses or statements related to the prompt_id.

generated: This seems to be a binary indicator, possibly distinguishing between human-generated and machine-generated texts.

To better understand the dataset, I will perform a preliminary analysis including:

Basic statistics for numerical columns.

Distribution of the prompt_id and generated columns.

A brief textual analysis of the text column.

Let's proceed with this analysis.

Here are the findings from the preliminary analysis:

Numerical Statistics:

There are 212 entries in the dataset.

The prompt_id column has values ranging from 0 to 1, which suggests it might be a binary category or a category with two distinct values.

The generated column is also binary, with values 0 and 1.

Distribution of prompt_id:

There are more entries with prompt_id 0 (140 entries) compared to prompt_id 1 (72 entries).

Distribution of generated:

A significant majority of the entries (184 out of 212) have a generated value of 1, indicating they might be machine-generated texts. Only 28 entries have a generated value of 0, which could indicate human-generated texts.

Text Length Statistics:

Count: There are 212 text entries.

Mean: On average, each text entry contains about 5,348 characters.

Standard Deviation: There is a high standard deviation (about 8,081), suggesting significant variation in text lengths.

Min/Max: The shortest text is 319 characters long, and the longest is 53,301 characters.

Quartiles:

25% of the texts are shorter than 1,905 characters.

50% (median) are shorter than 2,861 characters.

75% are shorter than 5,125 characters.

Let's start by exploring the data. We'll load each of the provided files (`train.csv`, `test.csv`, `train_prompts.csv`) and take a look at their content and structure.

The data files have the following structures:

1. Train Data (`train.csv`): Contains four columns:
   - `id`: Unique identifier for each record.
   - `prompt_id`: Identifier linking to the essay prompt.
   - `text`: The essay text.
   - `generated`: A binary flag indicating if the essay was generated by an AI (1) or written by a human (0).

2. Test Data (`test.csv`): Similar to the train data but without the `generated` column. It contains:
   - `id`: Unique identifier for each record.
   - `prompt_id`: Identifier linking to the essay prompt.
   - `text`: The essay text.

3. Train Prompts (`train_prompts.csv`): Contains the prompts used for the essays:
   - `prompt_id`: The identifier for each prompt.
   - `prompt_name`: Name of the prompt.
   - `instructions`: Instructions related to the prompt.
   - `source_text`: Additional text or context for the prompt.

4. Submission Example (`submission.csv`): Shows the expected format of the submission file, with only the `generated` column indicating the predictions.

Next steps:

1. Data Preprocessing: We'll join the `train_data` with `train_prompts` on `prompt_id` to include prompt details in the training data. We'll do the same for the `test_data`.
2. Feature Engineering: Extract features from the `text` and possibly from the prompt details.
3. Model Building and Evaluation: Train a classifier and evaluate it using the roc_auc metric.
4. Prepare the Submission: Use the trained model to make predictions on the `test_data` and format the submission file as per the example.

Let's start with data preprocessing.

The train and test datasets have been successfully merged with the prompt details. Each record now includes the essay text, prompt name, instructions, and source text, providing a more comprehensive context for the essays.

Next, we'll proceed with feature engineering. This involves creating features that can help distinguish between human-written and AI-generated essays. Some potential features to consider are:

1. Text Length: Length of the essay.
2. Word Count: Number of words in the essay.
3. Average Word Length: Average length of words in the essay.
4. Lexical Diversity: Variety of words used in the essay.
5. Readability Scores: Such as Flesch reading ease, which can indicate the complexity of the text.
6. Sentiment Analysis: The overall sentiment of the essay.
7. Prompt Relevance: Measures how closely the essay text is related to the prompt.

For this task, we'll focus on some of these basic features to start with. After extracting these features, we'll train a model and evaluate it using the roc_auc metric.

The model with random forest achieved a ROC AUC score of approximately 0.997 on the validation set, which indicates a very high level of predictive accuracy. However, this extremely high score might suggest overfitting, especially since we only used basic text features. It's important to be cautious and consider additional validation or more sophisticated feature engineering in a real-world scenario.

Next, we will prepare the submission file. We'll use the trained model to make predictions on the test dataset and format the submission according to the provided example.

The final step is to save this submission file as a CSV, following the guideline to not include the index.

In case of any feedback, Connect me on LinkedIn: https://www.linkedin.com/in/iam-manoj/