Decoding and Explainability ML and LLMs

The provided code is a Python script that analyzes the feature importance in a dataset using various data analysis and machine learning techniques. Below is a detailed report of the code's functionality and processes:

Libraries and Modules:

The code imports several essential libraries and modules such as re, numpy, pandas, csv, and various modules from the sklearn and matplotlib.

It includes custom modules lib.dataset and lib.chatGPT, containing classes and functions for dataset handling and accessing a ChatGPT model.

Main Function:

The main function serves as the entry point of the script. It performs the following operations:

1. Defines the paths for the dataset and the result file.
2. Instantiates a ChatGPT object.
3. Loads the dataset from the provided path and preprocesses it. Categorical columns are encoded as numerical codes.
4. Prepares a prompt for the ChatGPT model and generates a response based on a subset of the dataset, extracting features from the response.
5. Calculates feature importance using various methods such as Information Gain, Correlation, and Principal Component Analysis (PCA).
6. Writes the obtained results to a CSV file and analyzes the importance of features based on the calculated scores.
7. Finally, it visualizes the feature importance scores using a bar plot.

Feature Importance Methods:

The code implements three feature selection methods: Information Gain (IG), Correlation, and PCA. It then finds the intersection and union of important features identified by these methods.

Results and Visualization:

The script computes the importance of each feature and provides a visualization of feature importance scores using a bar plot. The bar plot displays the scores for each feature obtained from the different feature selection methods.

Handling Exceptions:

The code includes error handling for potential issues such as file not found and NotImplementedError.

Prompt engineering:

- **Role Assignment and Specification**

  Role Definition: The prompt assigns the role of “system,” establishing the context and purpose of the following communication.
  Instructional Content

  Professional Context: The prompt sets the scene by framing the user (ChatGPT) as a “data scientist,” providing a context for the subsequent instructions.

  Data Format Specification: It specifies the format of the input data, delineating the entries as “feature:value” pairs for each feature, ultimately leading to the label. This establishes the scope and structure of the data that the model should consider.

- **Feature Importance Ranking Instructions**

  Importance Ranking: Explicitly instructs the model to rank all features based on their importance, requiring the model to assign importance weights to each feature. This provides a clear directive for the model’s response generation.

  Listing Significant Features: Further instructs the model to list the important features in a specified format once the analysis is completed, clarifying the expected output.

  Example Illustration: Provides an illustrative example of how the important features should be listed, ensuring that the expected response format is unambiguous to the model.

Final features scoring:

Individual feature importance scores are calculated by assigning descending numerical weights to each feature within a particular feature selection method. Accumulated weights across all methods are used to determine overall feature importance scores.

The total feature importance scores are computed and sorted, providing an overall ranking of features based on their cumulative weights. These scores are then normalized to generate percentage-based feature importance weights.

The final feature importance scores are visualized using a bar plot to provide a clear representation of the importance levels of individual features.