User Guide for Bug_Reports.ipynb

Introduction

The Bug_Reports.ipynb script serves as a processor of bug reports through machine learning operations to produce analysis results. This application implements Natural Language Processing (NLP) to analyze bug descriptions for classification purposes and conducts model evaluation on multiple classification strategies.

Prerequisites

Make sure you install all needed dependencies before starting the script execution. Please find the complete list of necessary packages together with their acceptable versions in requirements.pdf.

Dataset Requirements

- The program needs input as CSV files which hold bug report information.
- The minimum dataset columns needed are Body and Labels in addition to other required fields.
    - Body: The textual description of the bug.
    - class or Labels: The category or type of bug.

Steps to Run the Script

1. **Clone the GitHub Repository**
2. git clone <repository_url>
3. cd <repository_folder>
4. **Open Jupyter Notebook**
5. jupyter notebook
6. **Load the Notebook**
    - Open Bug_Reports.ipynb in Jupyter Notebook.
    - Ensure the dataset files are correctly placed in the expected directory.
7. **Run the Cells in Order**

Users must follow steps for running the script that involve executing cells in order to perform text preprocessing before model training and results evaluation.

Functionality Overview

1. Data Preprocessing

- Loads bug report data from CSV files.
- The program executes text preprocessing operations which include removing special characters and converting data to lowercase format and stopping word removal from text.

2. Feature Extraction

The text data transformation into numerical features happens through TF-IDF vectorization processes.

3. Model Training and Evaluation

- Trains a Naive Bayes classifier and a Random Forest model.
- Evaluates models using accuracy, precision, recall, and F1-score.
- Conducts statistical significance testing (paired t-test) to compare model performances.

4. Results and Visualization

- The system presents tabular results which show the outcomes of classification.
- The system demonstrates performance details through visual representations.

Expected Output

- A classification report with accuracy, precision, recall, and F1-score for each model.
- Statistical test results to compare model performance.
- Graphical representation of accuracy distributions.
- A classification_results.csv file containing detailed results.

Troubleshooting

- **Dataset not found**: Ensure the dataset files are correctly placed in the specified directory.
- **Missing dependencies**: Install the required packages using:
- pip install -r requirements.txt
- **Performance issues**: Reduce the dataset size or limit the number of features in TF-IDF vectorization.

Contact

For any issues or improvements, please raise an issue on the GitHub repository or contact the project maintainer.