News Article Categorization Project

Objective

Develop a machine learning model to classify news articles into predefined categories accurately. This task simulates the process of identifying various use cases of generative AI in textual data and aligns well with the focus on understanding AI deployment across different sectors.

Timeline

Please complete the assessment in no more than 90 minutes. We know this is an open-ended project, and you could take many different approaches, some of which may be more time-consuming than others. A standardized duration helps ensure that we can fairly evaluate all candidates.

Work Product

You may use Google Colab or publish your work to a repo. We prefer the repo to showcase your work and organizational skills, but we need you to stay within the 90-minute window.

Dataset

Utilize one of the following datasets available on Kaggle:
News Category Dataset
BBC News Classification Dataset

Environment Setup

Tools Required: Python and Jupyter Notebook.
Python Libraries: pandas, scikit-learn, NLTK or spaCy, matplotlib or seaborn (for visualization).
Link to the Google Colab: News Classification Project.ipynb

Task Requirements

a. Data Preprocessing

- Load the dataset using pandas.
- Handle missing values, if any, and remove duplicate entries.
- Normalize the text (convert to lowercase, remove punctuation and numbers).
- Tokenize the text and remove stopwords.

b. Feature Engineering

- Convert the cleaned text data into numerical features using one of the following techniques:
    - TF-IDF Vectorization.
    - Countvectorizer
    - Consider experimenting with word embeddings for advanced feature extraction.

c. Model Building

- Split the dataset into training and testing sets.
- Build at least two different models:
    - A simpler model (e.g., Logistic Regression or Naive Bayes).
    - A more complex model (e.g., Random Forest or a neural network model).
- Train these models on the training data.

d. Model Evaluation

- Evaluate the models using accuracy, precision, recall, and F1-score metrics.
- Use a confusion matrix to provide a visual understanding of the model performance.

Deliverables

- A Jupyter notebook with implemented code, detailed comments, and visualizations.
- A comprehensive report summarizing the methodology, results, and insights. The report should discuss:
    - Steps were taken in data preprocessing and feature engineering.
    - Comparison between the models used.
    - Analysis of model performance with supporting visualizations.
    - Conclusions and recommendations for model deployment.

Evaluation Criteria

- **Correctness and Completeness**: All project stages (from data loading to model evaluation) should be logically and correctly implemented.
- **Model Performance and Metrics**: Selection of appropriate metrics for evaluating model performance and thoroughly analyzing results.
- **Quality of Code**: The code should be clean, well-documented, and easily understood.
- **Depth of Analysis in the Report**: The report should provide deep insights into the project and communicate the steps, findings, and recommendations.

Final Notes

- You are encouraged to spend about 90 minutes on this assignment:
    - If you run out of time, please document what you would do with more time
    - If you complete the tasks and have more time, please feel free to provide additional analysis and showcase different models and their trade-offs.
- This is your chance to show off; please don’t limit yourself to what’s asked in this Google document or template; these are just the minimum requirements of what we want to see.
- The Jupyter Notebook template is a suggestion to make things easier, but you will likely add more cells, especially to do more analysis and write clean code.
- If you have any questions about technical difficulties with the notebook, please don't hesitate to let us know; that information will not be considered in the evaluation process.

- However, you are responsible for independently handling any bugs in your code to make sure it at least works within the platform you coded on