Natural Language Processing Project

.

Requirements

Project Requirements for NLP Classification Tasks

1. Data Collection

- **Dataset Acquisition**: Download the provided dataset or acquire a similar dataset from a reliable source like Kaggle, UCI Machine Learning Repository, or public datasets related to your classification task.

- **Data Storage:** Store the dataset in a structured format (CSV, JSON, etc.) that can be easily loaded for analysis and model training.

2. Data Exploration and Visualization

- **Data Inspection:** Load the data and check for basic properties:
  - Number of records, types of variables, and missing values.
  - Perform basic descriptive statistics on numerical columns (e.g., counts, means, medians).
- **Visualization**: Create visualizations to understand the data better:
  - **Text Length Distribution:** Plot the distribution of text length (number of characters or words).
  - **Label Distribution:** If applicable, visualize the distribution of the different classes in the dataset (e.g., categories in news classification, sentiment types in mental health classification).
  - **Word Clouds:** Visualize the most frequent words in the dataset using word clouds.
  - **Class Balance:** Check for class imbalance issues (e.g., certain labels may have more data than others).

3. Data Preprocessing

- **Text Cleaning:**
  - Remove unnecessary characters (punctuation, special symbols, extra whitespaces, etc.).
  - Convert all text to lowercase.
  - Tokenization: Split the text into words or tokens.
  - Remove stop words (optional based on the project).
  - Lemmatization/Stemming: Convert words to their root form.
  - Handle contractions (e.g., "don't" to "do not").

- **Missing Data Handling**: Address missing values if any (e.g., filling, removing).
- **Outlier Detection**: For numerical data (if applicable), identify and handle outliers.

4. Feature Engineering

- **Feature Creation:** creating new features of transforming existing features using domain knowledge of the data, that make machine learning algorithm work better
  - length of documents
  - average word size within a document
  - use of punctuation in the text
  - capitalization of words in a document
  - number of digits in document
  - …
- **Text Representation:** Choose a suitable representation for text data:
  - Bag of Words (BoW): Simple representation based on word counts.
  - TF-IDF: Term Frequency-Inverse Document Frequency, to highlight important words.
  - Word Embeddings: Pre-trained word embeddings (e.g., Word2Vec, GloVe, FastText) can be used for better semantic representation.
  - N-grams: Optionally, include bigrams or trigrams to capture more context.
- **Feature Scaling**: For numerical features (if any), scale the features to normalize them.

5. Model Selection and Training

- **Model Choice:** Choose at least three of the following models:
  - Logistic Regression
  - Naive Bayes
  - Support Vector Machines (SVM)
  - Random Forest
  - Neural Networks (e.g., LSTM, GRU, CNN for text classification)
  - Transformer-based models (e.g., BERT, GPT-2, RoBERTa) for state-of-the-art performance
- **Model Evaluation**: Split the data into training and testing sets (e.g., 80-20 split or 10-fold cross-validation).
  - Use evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
  - In case of class imbalance, use additional metrics like ROC AUC or Precision-Recall AUC.
- Important Note (Using Deep Learning Models, Transformers, and Cross Validation with k fold are Bonus If you want).

6. Hyperparameter Tuning

- Use techniques like Grid Search or Randomized Search to tune the hyperparameters of the chosen model.
- Track metrics during training to avoid overfitting (e.g., using validation sets or cross-validation).

7. Model Evaluation and Analysis

- Confusion Matrix: Visualize the confusion matrix to evaluate the true positive, false positive, true negative, and false negative rates.
- Classification Report: Provide a classification report with precision, recall, and F1-score for each class.
- Model Interpretation: Analyze which features or words contribute most to the model's decision-making (using feature importance or attention mechanisms in deep models). (Bonus)

8. streamlit or tkinter or any ui to take input and generate output **(Bonus)**

9. Reporting and Presentation

- **Documentation:** Write a detailed report that includes:
  - An introduction to the problem and dataset.
  - Data exploration and visualizations.
  - Preprocessing and feature engineering techniques used.
  - Model selection and training details.
  - Hyperparameter tuning and evaluation metrics.
  - A comparison of different models (if applicable).
  - Conclusion and suggestions for future improvements.
- **Presentation**: Prepare a presentation (slides or poster) summarizing the findings and methodologies. Be ready to discuss challenges faced, insights gained, and model performance.