Detailed report

Approach

1. Data preprocessing

Loading and processing data: To kick off the project, data was loaded from a JSONL (JSON Lines) file. Each line in the file contained a JSON object representing a news article. This format allows for efficient line-by-line parsing and processing, which was crucial given the size of the dataset.

- **Data extraction**: The JSON data was extracted to obtain user and assistant messages. Specifically, the 'user' messages were used as the text data for classification, while the 'assistant' messages were analyzed for labels indicating whether an article was "Valid" or "Invalid".
- **Text cleaning**:
    - **Removing URLs**: URLs were removed using regular expressions to eliminate irrelevant web links from the text, which could introduce bias or distract from the actual content of the news articles.
    - **Handling mentions and hashtags**: Mentions and hashtags were removed because they were specific to social media platforms and did not contribute to the textual content of the articles.
    - **Removing non-alphanumeric characters**: Special characters and punctuation were stripped out to normalize the text and focus on the essential content.
    - **Whitespace management**: Multiple spaces were consolidated, and leading/trailing spaces were removed to ensure consistency in the text data.

Data splitting: The data was divided into training and validation sets to evaluate the model's performance effectively. This process involved:

- **Training set**: Used to train the model, allowing it to learn from the examples.
- **Validation set**: Used to evaluate the model's performance during training and tune hyperparameters.

2. Model training

Text tokenization: The text data was tokenized using Keras's Tokenizer, which converted text into sequences of integers where each integer represented a word in the dataset.

- **Tokenization process**:
    - **Text to sequences**: Each text was converted into sequences of integers.
    - **Padding sequences**: Sequences were padded to a uniform length, ensuring that all input data had the same shape for the model.

Building the model: An LSTM (Long Short-Term Memory) model was chosen due to its effectiveness in handling sequential data like text.

- **Model architecture**:
    - **Embedding layer**: Transformed tokenized words into dense vectors of fixed size.

- **SpatialDropout1D layer**: Applied dropout to the embedding layer to prevent overfitting.
- **Bidirectional LSTM layer**: Processed sequences in both forward and backward directions, capturing context from both past and future tokens.
- **Dense output layer**: Produced binary classification results using a sigmoid activation function.

Training the model: The model was trained with:

- **Early stopping**: A technique that halted training when the model's performance on the validation set no longer improved, preventing overfitting and ensuring optimal performance.

Other models considered:

1. **Naive Bayes**:
    - **Approach**: A probabilistic model based on Bayes' Theorem because it is simple and effective for text classification tasks.
    - **Challenge**: Assumed feature independence, which often does not hold for text data where word dependencies exist.
2. **Support Vector Machines (SVM)**:
    - **Approach**: A supervised learning model that finds the hyperplane maximizing the margin between classes because it is effective for classification tasks with high-dimensional data.
    - **Challenge**: May struggle with very large datasets and text data due to the need for extensive feature engineering.

Final choice: The LSTM model was selected due to its balance of complexity and effectiveness for sequential text data.

3. Model evaluation

Training history: Training history plots for accuracy and loss over epochs were generated to monitor model performance and convergence.

- **Accuracy and loss plots**:
    - **Accuracy plot**: Showed how well the model's predictions matched the true labels over time.
    - **Loss plot**: Illustrated how the model's prediction error changed during training.

Performance metrics: The model's performance was assessed using:

- **Accuracy**: Overall proportion of correct predictions.
- **Precision**: The ratio of correctly predicted positive cases to the total predicted positives.
- **Recall**: The ratio of correctly predicted positive cases to the actual positives.
- **F1-score**: The harmonic mean of precision and recall, providing a balanced measure.

4. Explainability

SHAP (shapley Additive explanations): SHAP values were used to explain model predictions by quantifying the contribution of each word to the final decision.

- **Explainability process**:
    - **SHAP values calculation**: Estimated how each word influenced the prediction.
    - **Generating explanations**: Provided human-interpretable explanations for model decisions.

Models and performance

Model performance: The LSTM model achieved an accuracy of 0.83 on the validation set, indicating that the model performed well in classifying news articles as valid or invalid.

Evaluation summary:

- **Accuracy**: 0.83, demonstrating the model's effectiveness.
- **Precision, Recall, and F1-Score**: Metrics were used to assess the balance between correctly identifying positive cases and avoiding false positives/negatives.

Challenges

Data cleaning challenges:

- **Handling diverse text forms**: The text data came from various sources and included a range of formatting issues. This required:
    - **Robust preprocessing techniques**: To clean the text effectively and ensure the model was trained on high-quality data.

Model overfitting:

- **Overfitting management**:
    - **Balancing complexity**: Ensuring that the model was complex enough to learn from the data but not so complex that it overfitted to the training set.
    - **Early stopping and regularization**: Used to manage overfitting by monitoring validation performance and applying dropout techniques.

Explainability challenges:

- **Understanding SHAP values**:
    - **Complex interpretations**: Translating SHAP values into meaningful explanations required a deep understanding of both the model and the SHAP methodology.