PDAN8412 POE PART 1

Makabongwe Lwethu Sibisi

ST10145439

Table of Contents

QUESTION 1 2

Data Needed for textual data analysis with an RNN 2
Data Quality Considerations & Common Pitfalls: 2

QUESTION 2 3

EDA 3
Feature Selection 4
Train model 4
Interpret and evaluate model 5
Report 6

QUESTION 3 7

QUESTION 4 7

QUESTION 5 7

Question 1

Data Needed for textual data analysis with an RNN

Textual data analysis for a Recurrent Neural Network (RNN) begins with collecting raw text, such as sentences, paragraphs, or documents. This text is then tokenized into smaller units, typically at the word, character, or sub word level. Each token is converted into a numerical representation, such as one-hot encoding which allows the RNN to process the information. Sequences are often padded or truncated to a fixed length to ensure consistency during training. For supervised tasks, corresponding labels are required to guide the learning process and optionally, additional metadata like author information or topic can be included to provide more context. (GeeksforGeeks, 2025)

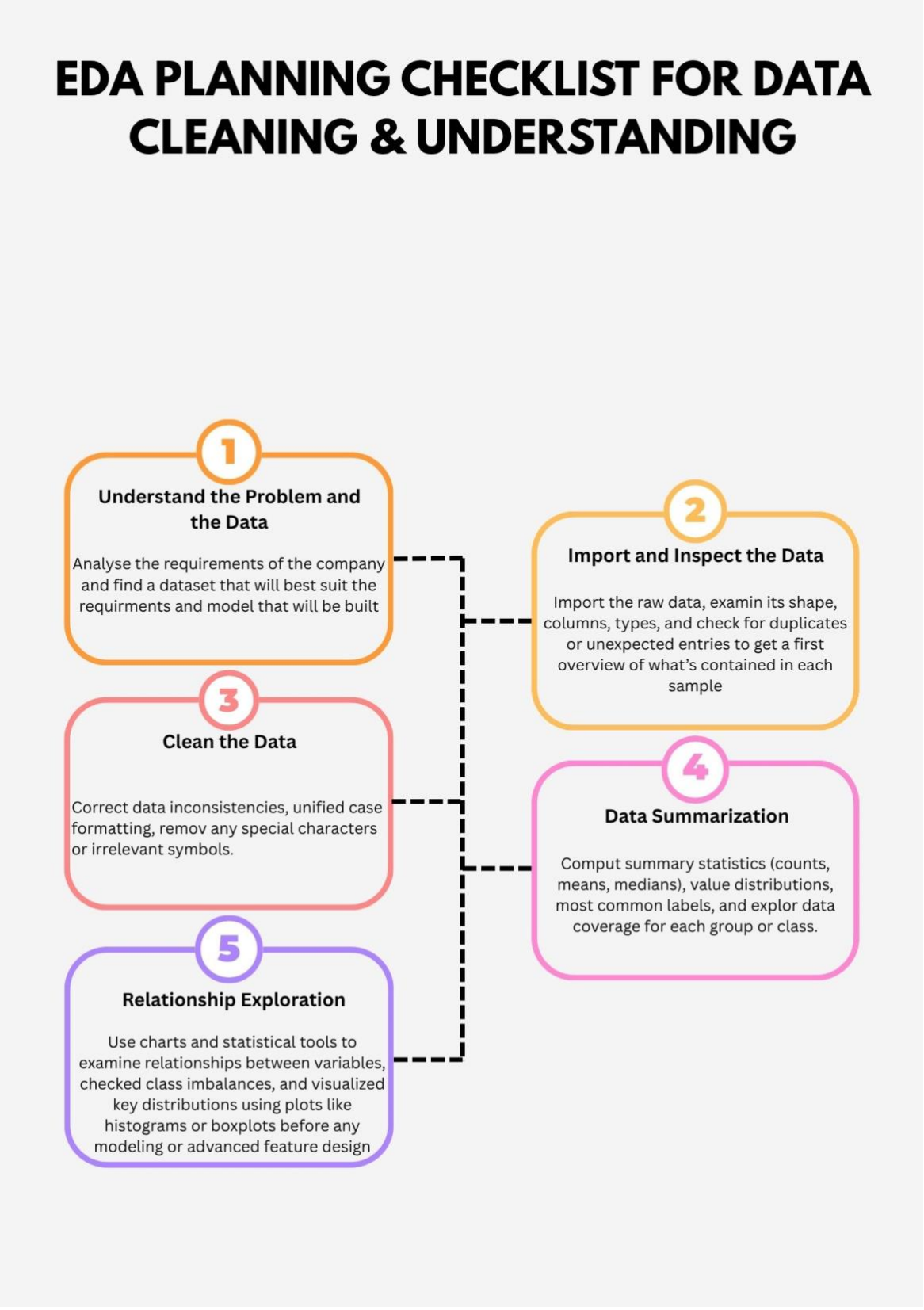
In the context of a Long Short-Term Memory (LSTM) Recurrent Neural Network designed to categorize a user's input to guess the author of a quote, the data required will include tokenized sequences of the quotes and labelled data indicating the author of each quote. Without labelled data, the model cannot learn which features are associated with each author, making accurate categorization impossible. Using supervised, labelled data simplifies the learning process and allows the LSTM to effectively capture stylistic and linguistic patterns unique to each author, improving its predictive performance. This makes the Spooky Authors dataset from Kaggle particularly suitable, as it provides a large collection of quotes explicitly labelled by author. The dataset includes distinct examples from multiple authors, ensuring that the model can learn author-specific patterns and make accurate predictions on new, unseen text inputs.. (T.J.J, 2020)

Data Quality Considerations & Common Pitfalls:

When preparing textual data for an RNN, the quality of the data is just as important as its quantity. Common issues include inconsistent formatting, spelling errors, excessive punctuation, and the presence of irrelevant or noisy text, all of which can reduce model performance. Imbalanced datasets, where certain authors or categories dominate, can bias the model and lead to poor generalization. It's also important to ensure that sequences are appropriately pre-processed, tokenized correctly, padded consistently. It must also be cleaned of unnecessary symbols. Another frequent pitfall is using unlabelled data for supervised tasks, which prevents the network from learning meaningful patterns associated with specific outputs. Finally, overly small datasets can cause overfitting, while extremely large datasets without proper preprocessing can introduce noise and slow training. Careful attention to these factors ensures that the RNN receives clean, representative, and properly labelled data, which is crucial for reliable learning and accurate predictions. (A.Lones, 2024)

Question 2

EDA



Feature Selection

For this dataset, explicit feature selection will not be required, as all the information it contains being quotes and their corresponding authors, is already directly relevant to the task of author classification. However, in general, feature selection could involve identifying and using only the most informative parts of the text, such as specific words, phrases, or stylistic markers (e.g., sentence length, punctuation patterns, use of adjectives or adverbs) that help distinguish between authors. Techniques such as TF-IDF weighting, n-gram extraction, or part-of-speech analysis can also be applied to reduce noise and emphasize features that contribute most to predictive performance in text classification tasks.

Train model

I will train the model using the following planned steps and hyperparameters:

1. **Prepare Data Inputs:**
   I will tokenize and pad text sequences to a fixed length, and encode target labels into numeric format suitable for classification.
2. **Define Model Architecture:**
   I will create an embedding layer (e.g., 128 dimensions), followed by a bidirectional LSTM layer (e.g., 64 units), possibly a dropout layer (e.g., rate of 0.5), and a dense output layer with softmax activation for multiclass classification.
3. **Set Training Hyperparameters:**
   - **Batch Size:** Set to 64 to balance memory usage and training stability.
   - **Epochs:** Allow the model to train for up to 30 epochs, with early stopping.
   - **Optimizer:** Use Adam with a learning rate of 0.001 for efficient convergence.
   - **Validation Split:** Reserve 20% of the data for validation during training.
4. **Implement Training Callbacks:**
   - **EarlyStopping:** Monitor validation loss to halt training if it doesn't improve for 5 epochs, restoring best weights.
   - **ReduceLROnPlateau:** Lower the learning rate by a factor (e.g., 0.2) if validation loss stops improving.
5. **Monitor Model Performance:**
   I will track accuracy and loss on both training and validation sets after each epoch to assess performance and detect overfitting.

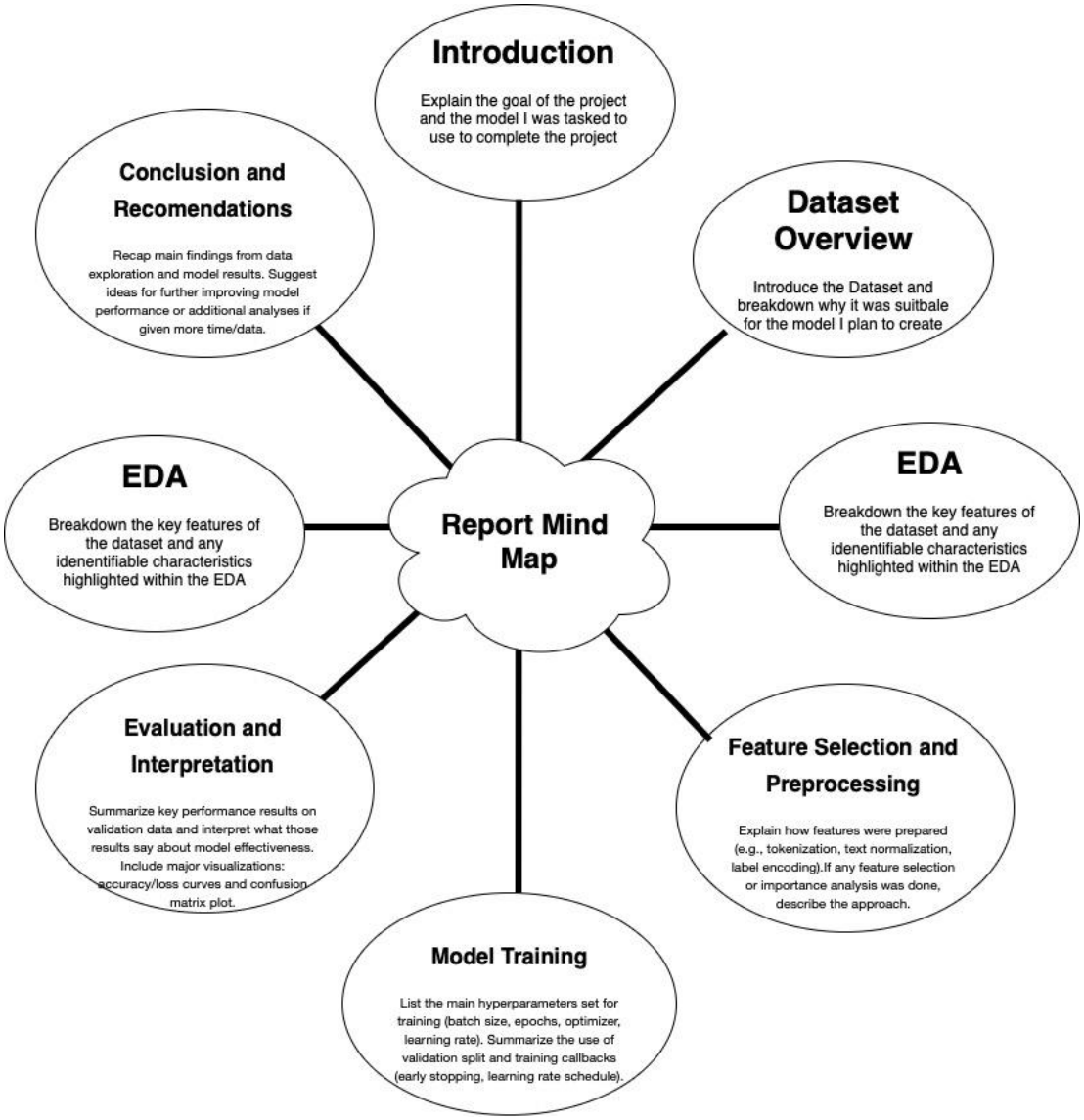
6. **Evaluate and Tune:**
   After training, I will review validation accuracy, loss curves, and classification metrics. If needed, I will adjust hyperparameters such as LSTM units, dropout rates, batch size, and learning rate for improved results.
7. **Document Best Settings:**
   I will record the final set of hyperparameters that yielded optimal validation performance, ensuring reproducibility and transparency of the model training process.

Interpret and evaluate model

I will use the following evaluation metrics to interpret the results of my model:

1. **Accuracy:**
   Measures the proportion of correct predictions out of all predictions, providing an overall assessment of model performance.
2. **Precision, Recall, and F1-score:**
   These metrics are reported for each class using a classification report. Precision measures the correctness of positive predictions, recall measures the ability to find all positive instances, and F1-score balances both for a comprehensive view.
3. **Confusion Matrix:**
   A confusion matrix is generated to visualize the number of correct and incorrect predictions for each class, helping to identify specific areas where the model may be making errors

Report


Report Mind Map
Introduction
Explain the goal of the project and the model I was tasked to use to complete the project
Dataset Overview
Introduce the Dataset and breakdown why it was suitbale for the model I plan to create
EDA
Breakdown the key features of the dataset and any idenentifiable characteristics highlighted within the EDA
Feature Selection and Preprocessing
Explain how features were prepared (e.g., tokenization, text normalization, label encoding).If any feature selection or importance analysis was done, describe the approach.
Model Training
List the main hyperparameters set for training (batch size, epochs, optimizer, learning rate). Summarize the use of validation split and training callbacks (early stopping, learning rate schedule).
Evaluation and Interpretation
Summarize key performance results on validation data and interpret what those results say about model effectiveness. Include major visualizations: accuracy/loss curves and confusion matrix plot.
EDA
Breakdown the key features of the dataset and any idenentifiable characteristics highlighted within the EDA
Conclusion and Recomendations
Recap main findings from data exploration and model results. Suggest ideas for further improving model performance or additional analyses if given more time/data.

Question 3

Please find Reponses within the github repo. The following YouTube videos were used as reference for the code.

https://github.com/just-makab/PDAN8412_POE_PART_1.git

Question 4

https://github.com/just-makab/PDAN8412_POE_PART_1.git

Question 5

Please refer to the Report Document.

Bibliography

Keith, M., 2020. *Lwarn Exploratory Data Analysis(EDA) in Python*. [Online]
Available at:
https://www.youtube.com/playlist?list=PLe9UEU4oeAuV7RtCbL76hca5ELO_IELk4
[Accessed 25 April 2025].

Feely, C., 2021. *Feature Selection in Python | Machine Learning Basics | Boston Housing Data*. [Online]
Available at: https://www.youtube.com/watch?v=iJ5c-XoHPFo
[Accessed 25 April 2025].

The AI & DS Channel, 2022. *Medical insurance cost prediction using linear regression | Machine Learning Project 7*. [Online]
Available at: https://www.youtube.com/watch?v=iS_iI7btRXk
[Accessed 25 April 2025].

A.Lones, M., 2024. *Avoiding common machine learning pitfalls*. [Online]
Available at: https://www.sciencedirect.com/science/article/pii/S2666389924001880
[Accessed 1 October 2025].

GeeksforGeeks, 2025. *What is LSTM - Long Short Term Memory?*. [Online]
Available at: https://www.geeksforgeeks.org/deep-learning/deep-learning-introduction-to-long-short-term-memory/
[Accessed 1 October 2025].

T.J.J, R., 2020. *LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras*. [Online]
Available at: https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2
[Accessed 1 October 2025].