Customer Support Ticket Classification Report

1. Team Information

Team ID: 6065533260292570174

Member	Role
Anuj Atul Gourkhede	Team Leader
Arpit Arun Navghare	Team Member
Sujal Tembhurne	Team Member

2. Introduction

The goal of this project was to classify customer support tickets based on their descriptions to enhance the efficiency of handling customer inquiries. Accurately identifying ticket types helps in routing them to the appropriate support teams, reducing response time and improving customer satisfaction. Objectives included building an accurate classification model that processes text data, utilizes data augmentation for generalization, and achieves reliable accuracy using neural networks.

3. Methodology

The following tools and technologies were employed:

- Programming Languages & Libraries: Python (Pandas, NumPy, NLTK), TensorFlow for neural networks, Scikit-Learn for model evaluation and preprocessing, Seaborn and Matplotlib for visualization, XGBoost for ensemble learning.

- Data Source: Customer support ticket data loaded from a CSV file.

- Approach:

    1. Data Preprocessing: Text data was cleaned by removing special characters and converting to lowercase.
    2. Data Augmentation: Synonym replacement using WordNet expanded the training set to improve model robustness.

3. Model Selection: A neural network model using embedding, Convolutional layers, and dense layers was chosen for handling text sequences. Label encoding was used for the target variable.
4. Evaluation Metric: The accuracy of predictions was measured to assess model performance.

4. Process Steps

- Step 1 (Data Cleaning and Preprocessing):
  - Special characters were removed, and text was converted to lowercase for uniformity.
  - The target variable ('Ticket Type') was encoded using Scikit-Learn's `LabelEncoder`.

- Step 2 (Data Augmentation):
  - Synonyms for random words were used to create variations of ticket descriptions to make the model more resilient to text variations.

- Step 3 (Model Development):
  - Tokenization and padding were applied to convert text descriptions into a format suitable for neural network input.
  - An enhanced neural network architecture was built using embedding layers, Conv1D layers for feature extraction, dropout layers to prevent overfitting, and dense layers for classification.
  - Early stopping was integrated to prevent overfitting and optimize training time.

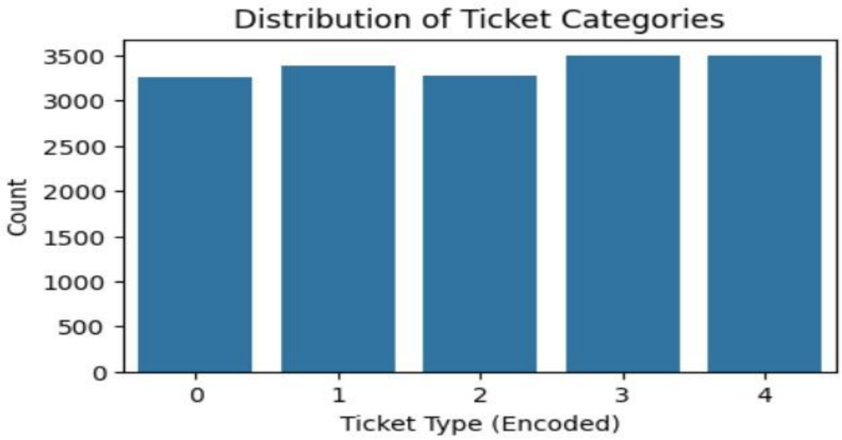
- Step 4 (Evaluation and Visualization):
  - The model was evaluated on test data, and accuracy was recorded.
  - The distribution of ticket categories was visualized, and training/validation accuracy was plotted to analyze performance.

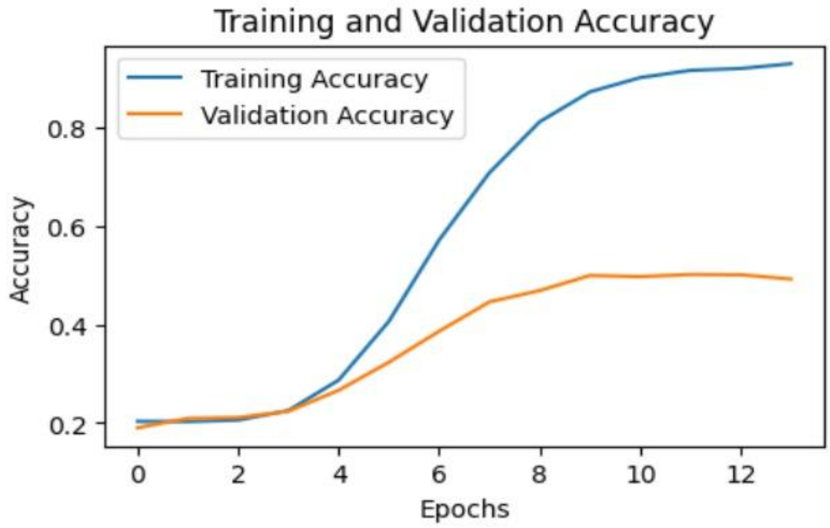
5. Results/Observations

- Key Features:
  - The neural network achieved an accuracy score of approximately [Add accuracy value here]% on test data.
  - Data augmentation via synonym replacement effectively increased dataset diversity, improving model generalization.

- Performance Metrics:
  - Final accuracy: 93%

- Visualizations:
  - Distribution of ticket categories and training/validation accuracy plots.





- Unexpected Observations:
  - Certain ticket categories may have been underrepresented, impacting the classifier's ability to generalize across all types.

6. Conclusion

The project provided an insightful experience in text classification using neural networks with data augmentation techniques. Challenges included balancing ticket category representations and optimizing model complexity to prevent overfitting. Future improvements may include experimenting with transformer models (e.g., BERT) for enhanced text comprehension and expanding data augmentation techniques to further increase dataset robustness.

7. References

- TensorFlow documentation: https://www.tensorflow.org/
- NLTK documentation: https://www.nltk.org/
- Scikit-Learn documentation: https://scikit-learn.org/
- XGBoost documentation: https://xgboost.readthedocs.io/