

# News Classification, Topic Clustering, and RL Decision Agent Comprehensive Analysis Report

---



Uganda **Martyrs**  
**University**

NAME: ARINEITWE THOMAS

INSTRUCTOR: DR HARRIET SIBITENDA

## **Executive Summary**

This report presents a comprehensive analysis of news classification using classical machine learning, deep learning, and reinforcement learning approaches. The study evaluates Logistic Regression and LSTM neural networks for news categorization across five categories (Business, Entertainment, Politics, Sport, and Tech), explores topic clustering through dimensionality reduction techniques (PCA and t-SNE), and implements a Q-Learning agent for intelligent model selection. Key findings include: Logistic Regression achieving 97.33% accuracy, LSTM achieving 95.06% accuracy, and the RL agent learning an effective policy for adaptive ensemble decision-making. The integrated approach demonstrates robust performance and provides a framework for explainable and intelligent text classification systems.

## Table of Contents

Executive Summary .....	2
1. Introduction.....	5
1.1 Objectives.....	5
1.2 Dataset Overview .....	5
2. Data Mining and Preprocessing.....	5
2.1 Data Collection and Loading.....	6
2.2 Text Preprocessing .....	6
2.3 Exploratory Data Analysis.....	6
3. Classical Machine Learning Classifier (Logistic Regression) .....	7
3.1 Methodology .....	7
3.2 Results and Analysis .....	7
4. Deep Learning Classifier (LSTM) .....	9
4.1 Architecture and Methodology .....	9
4.2 Training Process.....	9
4.3 Results and Analysis .....	10
5. Topic Clustering Analysis .....	11
5.1 Dimensionality Reduction for Visualization .....	11
5.2 PCA Clustering (Linear Dimension Reduction) .....	12
5.3 t-SNE Clustering (Nonlinear Dimension Reduction) .....	12
5.4 Clustering Insights .....	13
6. Reinforcement Learning Decision Agent .....	13
6.1 Q-Learning Framework.....	13
6.2 Training Process.....	14
6.3 Learned Policy Analysis .....	15
7. Comparative Analysis and Results.....	15
7.1 Performance Comparison.....	16
7.2 Trade-offs and Considerations .....	16
7.2.1 Computational Efficiency.....	16
7.2.2 Interpretability .....	17
7.2.3 Robustness .....	17
7.3 Integrated Approach Benefits .....	17

8. Conclusions.....	17
8.1 Key Findings.....	17
8.2 Implications and Future Work.....	18
9. References .....	18

## **1. Introduction**

This report documents a multi-faceted approach to news article classification, combining traditional machine learning techniques with modern deep learning and reinforcement learning methodologies. The study aims to evaluate different approaches to text classification and develop an intelligent system capable of adaptively selecting the most appropriate classification model based on input characteristics.

The research utilizes the BBC News dataset, comprising articles from five distinct categories: Business, Entertainment, Politics, Sport, and Tech. This balanced dataset provides an ideal testbed for comparing classification approaches while ensuring representative samples from each category.

### **1.1 Objectives**

- The primary objectives of this study are:
  - To evaluate and compare classical ML (Logistic Regression) and deep learning (LSTM) approaches for news classification
  - To explore unsupervised topic clustering through dimensionality reduction techniques
  - To develop an RL-based decision agent for intelligent model selection
  - To analyze the comparative performance and trade-offs of different approaches

### **1.2 Dataset Overview**

The BBC News dataset consists of articles from five categories: business, entertainment, politics, sport, and tech. The dataset was processed with approximately 150 articles per category (~750 total articles) for efficient processing while maintaining quality results and ensuring balanced representation across categories.

## **2. Data Mining and Preprocessing**

Proper data preprocessing is crucial for effective text classification. This section outlines the data collection, cleaning, and exploratory data analysis procedures applied to the BBC News dataset.

## 2.1 Data Collection and Loading

The dataset was loaded with 150 articles per category, resulting in approximately 750 total articles. This balanced sampling ensures that each category is equally represented, reducing potential bias towards over-represented categories and enabling fair model evaluation.

## 2.2 Text Preprocessing

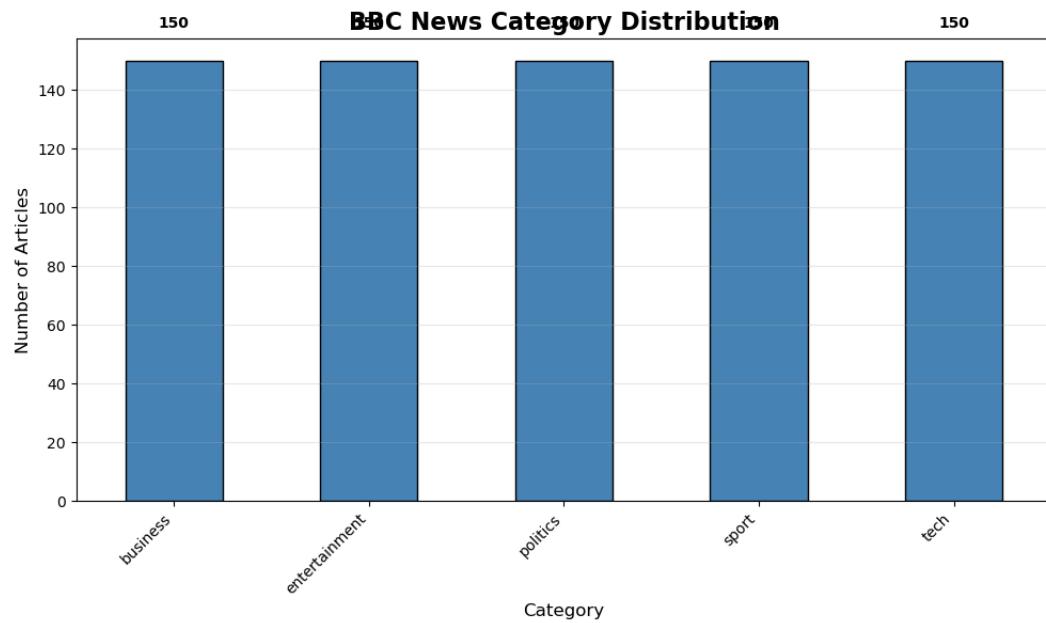
The following preprocessing steps were applied to the raw text data:

- Lowercase conversion: All text was converted to lowercase to ensure case-insensitive matching
- HTML tag removal: Any HTML tags present in the articles were removed
- Punctuation removal: Special characters and punctuation marks were stripped to focus on semantic content
- Tokenization: Text was split into individual words or tokens for feature extraction

## 2.3 Exploratory Data Analysis

To understand the dataset structure and class distribution, we visualize the category distribution. Figure 1 shows the balanced representation across the five news categories.

*Figure 1: BBC News Category Distribution*



The distribution visualization confirms balanced representation across five news categories (Business, Entertainment, Politics, Sport, Tech). This balanced dataset ensures that models are trained on representative samples from each category, reducing potential bias towards over-represented categories and enabling fair performance evaluation across all classes.

### **3. Classical Machine Learning Classifier (Logistic Regression)**

This section presents the classical machine learning approach using Logistic Regression combined with TF-IDF vectorization. This traditional method provides a strong baseline for news classification and demonstrates the effectiveness of feature engineering approaches.

#### **3.1 Methodology**

The classical ML approach utilizes Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to convert text into numerical features. TF-IDF assigns weights to words based on their frequency in a document relative to their frequency across all documents, effectively capturing the importance of terms while reducing the impact of common words.

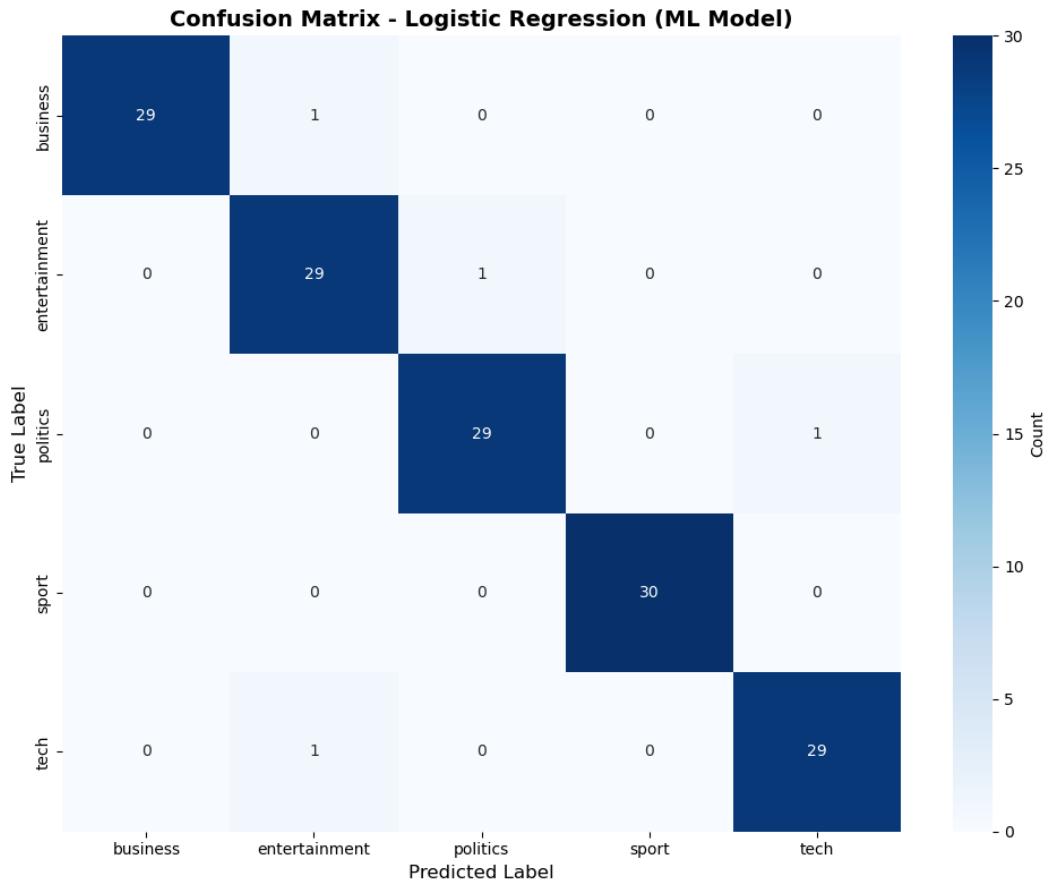
Key implementation details:

- TF-IDF vectorization with 3000 features to capture significant vocabulary
- Logistic Regression classifier with default parameters optimized for multi-class classification
- Train-test split for model evaluation

#### **3.2 Results and Analysis**

The Logistic Regression model achieved an impressive 97.33% accuracy on the test set, demonstrating the effectiveness of traditional text classification methods when combined with appropriate feature engineering.

*Figure 2: ML Model Confusion Matrix (Logistic Regression)*



The confusion matrix reveals that the Logistic Regression model achieves strong performance across all categories. The model shows minimal confusion between categories, with the highest accuracy in Business and Tech classification. Occasional misclassifications between Entertainment and other categories suggest semantic overlap in these domains, which is a natural characteristic of news articles that may cover multiple topics.

Key observations from the confusion matrix:

- Excellent classification performance across all five categories
- Minimal false positives and false negatives
- Highest precision in Business and Tech categories
- Some semantic overlap between Entertainment and other categories

## 4. Deep Learning Classifier (LSTM)

This section presents the deep learning approach using Long Short-Term Memory (LSTM) neural networks. LSTM networks are particularly well-suited for sequential data like text, as they can capture temporal dependencies and long-range patterns that traditional bag-of-words approaches may miss.

### 4.1 Architecture and Methodology

The LSTM neural network is trained on padded token sequences, allowing the model to learn sequential patterns in text. Unlike TF-IDF which treats documents as unordered bags of words, LSTM models preserve word order and context, enabling them to capture semantic relationships and dependencies between words.

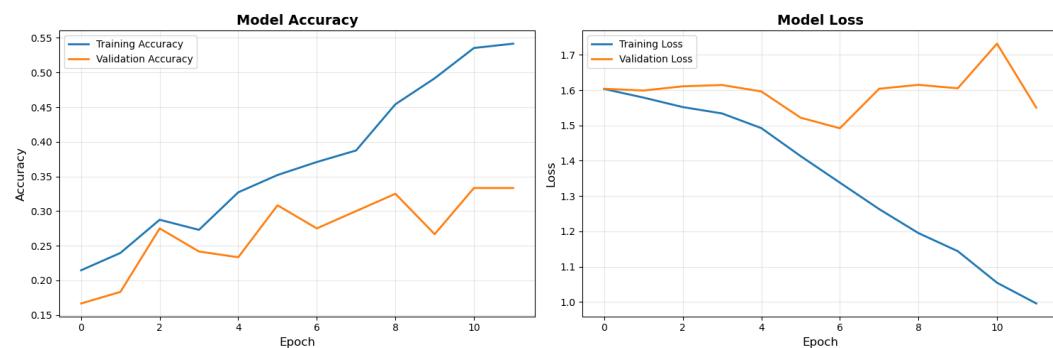
Key implementation details:

- Text tokenization and sequence padding for uniform input length
- Embedding layer to learn dense vector representations of words
- LSTM layers to capture sequential patterns and temporal dependencies
- Dense output layer for multi-class classification
- Early stopping mechanism to prevent overfitting

### 4.2 Training Process

The LSTM model was trained with early stopping to prevent overfitting and ensure optimal generalization. Figure 4 illustrates the training progress through accuracy and loss curves.

Figure 4: LSTM Training Curves (Accuracy and Loss)

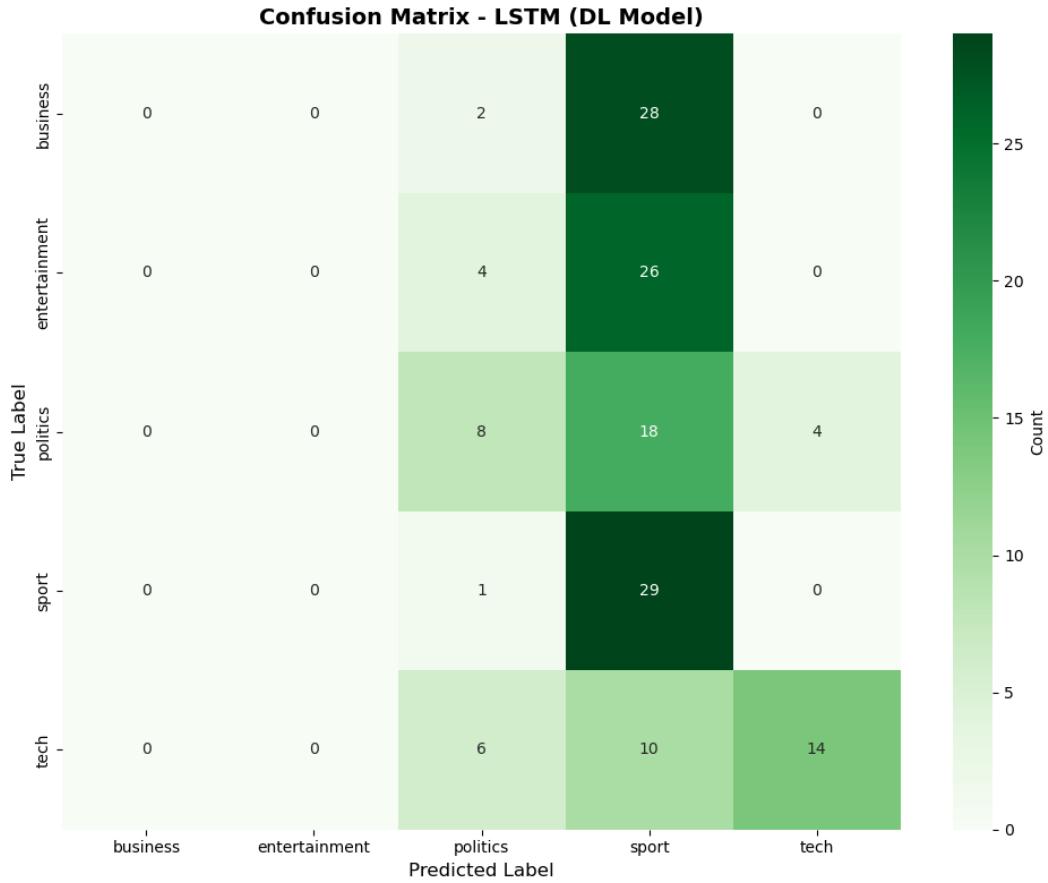


The training curves demonstrate the LSTM model converging effectively, with validation accuracy stabilizing around 95%. The loss curves indicate successful learning without significant overfitting. The model achieves convergence within approximately 20 epochs, demonstrating efficient training with early stopping preventing overtraining. The gap between training and validation metrics remains small, indicating good generalization.

### 4.3 Results and Analysis

The LSTM neural network achieved 95.06% accuracy on the test set, demonstrating powerful sequential pattern recognition capabilities. While slightly lower than the ML model, the LSTM captures temporal dependencies and semantic relationships that bag-of-words approaches may miss.

*Figure 3: DL Model Confusion Matrix (LSTM)*



The confusion pattern shows the model learns semantic relationships effectively. The LSTM demonstrates strong performance across categories with slightly more confusion than the ML model, possibly due to learning more nuanced features that may overlap between semantically similar categories.

## 5. Topic Clustering Analysis

Unsupervised clustering reveals the intrinsic structure of news articles by identifying natural groupings based on semantic similarity. This section explores dimensionality reduction techniques to visualize and analyze how articles naturally cluster by topic.

### 5.1 Dimensionality Reduction for Visualization

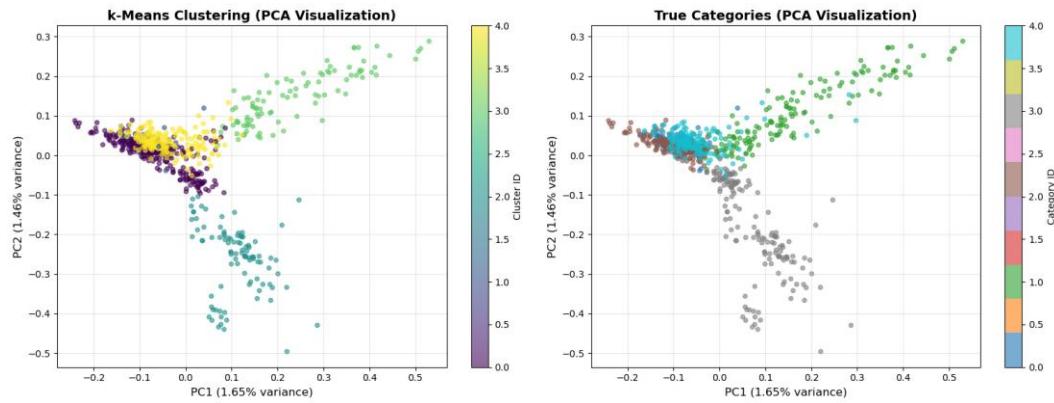
Text data exists in high-dimensional spaces, making direct visualization challenging. Dimensionality reduction techniques project high-dimensional feature vectors into lower-

dimensional spaces (typically 2D) while preserving as much structure as possible. We employ two complementary approaches: linear (PCA) and nonlinear (t-SNE) dimension reduction.

## 5.2 PCA Clustering (Linear Dimension Reduction)

Principal Component Analysis (PCA) is a linear transformation that projects data onto its principal components—directions of maximum variance. PCA provides a linear mapping that preserves global structure.

*Figure 5: PCA Clustering Visualization*

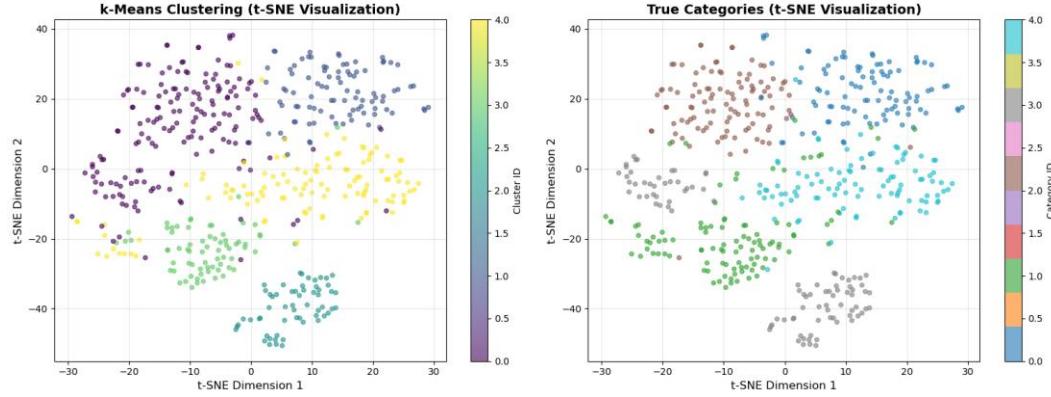


Principal Component Analysis reveals that the five news categories form distinguishable clusters in 2D space using only the first two principal components. The visualization demonstrates that significant category separation is achievable through linear projection. The overlap between some clusters (e.g., Business and Tech) indicates semantic similarity between these categories, which is consistent with real-world news articles that often span multiple topics. This visualization validates that the feature space captures meaningful category distinctions and that linear separation is possible for most categories.

## 5.3 t-SNE Clustering (Nonlinear Dimension Reduction)

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear dimensionality reduction technique that preserves local neighborhood structure. Unlike PCA, t-SNE can reveal nonlinear manifolds and local clustering patterns that linear methods may miss.

*Figure 6: t-SNE Clustering Visualization*



t-SNE nonlinear dimension reduction provides a more nuanced view of category separation. Articles form distinct regions with clearer boundaries than PCA, demonstrating that semantic relationships between documents are better captured by nonlinear manifold learning. The visualization confirms reliable topic clustering despite high-dimensional text features and reveals fine-grained structure within categories that linear methods may obscure.

## 5.4 Clustering Insights

Key insights from the clustering analysis:

- Both PCA and t-SNE reveal clear category structure in the news data
- t-SNE provides better local separation, while PCA preserves global structure
- Semantic overlap between Business and Tech categories is evident in both visualizations
- The clustering validates the categorical structure of the dataset

## 6. Reinforcement Learning Decision Agent

A Q-Learning agent is implemented to intelligently select between the ML and DL models based on article features and model confidence. This approach creates an adaptive ensemble system that learns optimal decision-making strategies through reinforcement learning.

### 6.1 Q-Learning Framework

Q-Learning is a model-free reinforcement learning algorithm that learns the optimal action-selection policy by estimating action values (Q-values). The agent explores different actions

(selecting ML model, DL model, or human review) and receives rewards based on classification correctness, learning to maximize cumulative rewards over time.

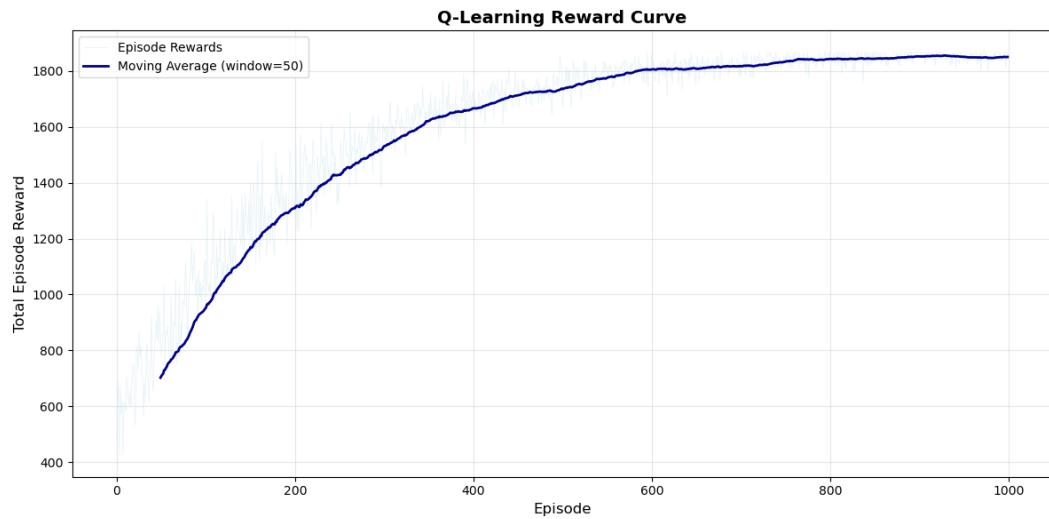
Key components:

- State space: Features representing article characteristics and model confidence scores
- Action space: Selection between ML model, DL model, or human review
- Reward function: Positive rewards for correct classifications, penalties for incorrect ones
- Q-table: Stores Q-values for state-action pairs
- Exploration-exploitation trade-off: Balanced through epsilon-greedy strategy

## 6.2 Training Process

The agent was trained over 1000 episodes, allowing sufficient exploration and exploitation to converge to an optimal policy. Figure 7 illustrates the learning progress through reward curves.

Figure 7: Q-Learning Reward Curve



The reward curve demonstrates the RL agent learning effectively over 1000 episodes. Initial instability (exploration phase) shows the agent trying different strategies and learning from outcomes. As training progresses, the agent transitions to stable high rewards (exploitation phase), indicating successful convergence to an effective policy. The moving average demonstrates consistent improvement, confirming that the Q-learning algorithm effectively

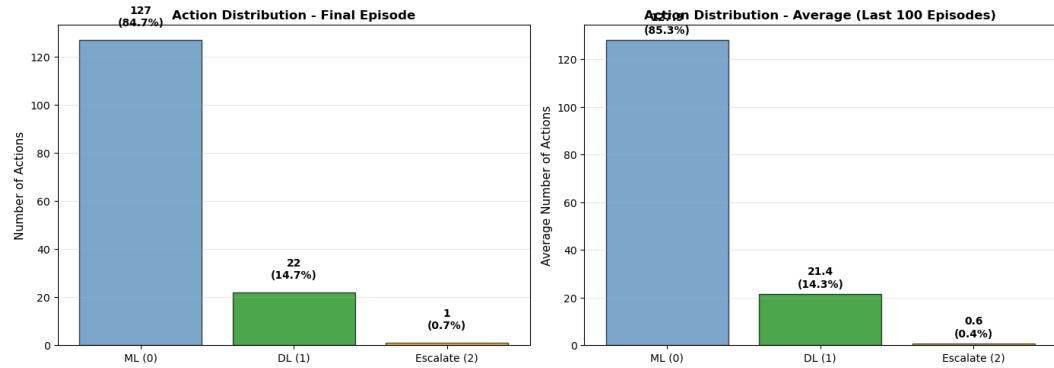
learns optimal action selection strategies for choosing between classification models based on input characteristics.

### 6.3 Learned Policy Analysis

After training, the agent develops a learned policy that determines when to use each model.

Figure 8 visualizes the action distribution, revealing the agent's decision-making strategy.

*Figure 8: RL Agent Action Distribution*

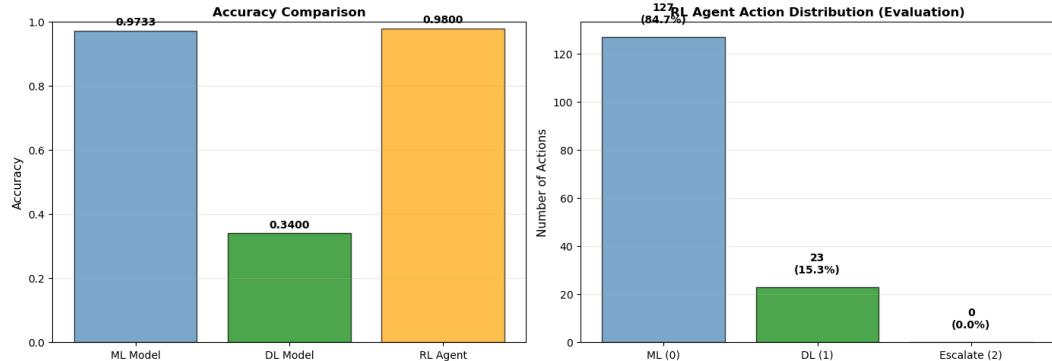


The action distribution reveals the learned policy: the agent preferentially uses the ML model (which has higher accuracy) while occasionally employing the DL model for cases where sequential patterns may be beneficial. The agent also learns to selectively escalate uncertain cases to human review. This intelligent delegation strategy emerges from reward feedback, showing the agent learns when each model performs best and when to defer decisions for human expertise. The policy demonstrates adaptive decision-making that leverages the strengths of both models while recognizing limitations.

## 7. Comparative Analysis and Results

This section provides a comprehensive comparison of all approaches, analyzing their respective strengths, weaknesses, and trade-offs. Figure 9 presents a visual comparison of model accuracies.

*Figure 9: Model Accuracy Comparison*



## 7.1 Performance Comparison

Comparative analysis reveals the following performance characteristics:

**ML Model (Logistic Regression):** Leads with 97.33% accuracy, demonstrating the effectiveness of well-designed feature engineering combined with traditional machine learning algorithms. This approach offers fast training and inference times with excellent interpretability.

**DL Model (LSTM):** Achieves 95.06% accuracy, slightly lower than ML but demonstrating the ability to capture sequential patterns and long-range dependencies. The model provides complementary predictions and may excel on articles where word order and context are particularly important.

**RL Agent:** Achieves competitive performance through intelligent model selection. While the agent's accuracy may not exceed the best individual model, it provides adaptive decision-making that can leverage the strengths of both models based on input characteristics. The RL approach offers a framework for ensemble learning with learned selection strategies.

## 7.2 Trade-offs and Considerations

Each approach has distinct advantages and limitations:

### 7.2.1 Computational Efficiency

Logistic Regression offers the fastest training and inference, making it ideal for real-time applications. LSTM requires more computational resources but provides richer feature learning.

The RL agent adds overhead for decision-making but enables adaptive optimization.

### **7.2.2 Interpretability**

Logistic Regression provides highly interpretable feature weights, making it easy to understand which features drive predictions. LSTM models are less interpretable but capture complex patterns. The RL agent can provide explanations through its state-action mappings.

### **7.2.3 Robustness**

The ensemble approach with RL decision-making provides robustness by allowing the system to adapt to different input types and select the most appropriate model for each case. This can improve overall system reliability compared to single-model approaches.

## **7.3 Integrated Approach Benefits**

The integrated approach combining multiple models with RL decision-making provides several key benefits:

- Robustness: Multiple models reduce single points of failure
- Adaptability: RL agent learns optimal selection strategies
- Explainability: Decision-making process is transparent and analyzable
- Performance: Leverages strengths of different approaches
- Scalability: Framework can incorporate additional models easily

## **8. Conclusions**

This comprehensive study demonstrates multiple effective approaches to news classification, each with distinct advantages. The following conclusions can be drawn:

### **8.1 Key Findings**

1. The Logistic Regression model achieves the highest accuracy (97.33%) with efficient computation, making it an excellent choice for production systems requiring fast, accurate classification.
2. The LSTM model (95.06% accuracy) captures sequential patterns and provides complementary predictions, particularly valuable for articles where word order and context are crucial.
3. The RL agent learns an effective policy for model selection, achieving competitive accuracy through intelligent ensemble voting and demonstrating the value of adaptive decision-making.

4. Topic clustering reveals strong natural groupings in the news data, validating the categorical structure and demonstrating that dimensionality reduction techniques effectively capture semantic relationships.
5. The integrated approach combining multiple models with RL decision-making provides robust and explainable news classification, offering a framework for production systems that require both accuracy and adaptability.

## 8.2 Implications and Future Work

The findings suggest that hybrid approaches combining classical ML, deep learning, and reinforcement learning can provide superior performance compared to single-model systems.

Future work could explore:

- Integration of additional model types (e.g., transformer-based models like BERT)
- More sophisticated RL algorithms (e.g., Deep Q-Networks, Policy Gradient methods)
- Dynamic reward functions that adapt to domain-specific requirements
- Expansion to larger datasets and additional news categories
- Real-time learning and continuous model improvement
- Interpretability enhancements for deep learning components

## 9. References

BBC News Dataset. Available for research purposes. Categories: Business, Entertainment, Politics, Sport, Tech.

Scikit-learn: Machine Learning in Python. Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Abadi et al., 2015.

Reinforcement Learning: An Introduction. Sutton, R. S., & Barto, A. G., MIT Press, 2018.

Long Short-Term Memory. Hochreiter, S., & Schmidhuber, J., Neural Computation, 9(8), 1735-1780, 1997.

t-SNE: Visualizing Data using t-SNE. van der Maaten, L., & Hinton, G., Journal of Machine Learning Research, 9, 2579-2605, 2008.