News Classification Project

ML/DL with Reinforcement Learning

Name: Arineitwe Thomas

Lecturer: Dr Sibitenda Harriet

Contents

1. Executive Summary

This project implements a comprehensive news article classification system that combines classical machine learning, deep learning, and reinforcement learning techniques. The system successfully classifies BBC news articles into five categories (business, entertainment, politics, sport, tech), discovers latent topics through clustering, and uses a reinforcement learning agent to intelligently decide which model to use or when to escalate to human review.

The project demonstrates the practical application of multiple machine learning paradigms in a real-world text classification scenario. Through systematic experimentation and evaluation, we achieved state-of-the-art performance by combining the strengths of different approaches.

Key Achievements:

- Classical ML Model (Logistic Regression): 98.88% accuracy with TF-IDF features
- Deep Learning Model (CNN): 95.06% accuracy using word embeddings and convolutional layers
- Reinforcement Learning Agent: 99.10% accuracy by intelligently selecting optimal model
- Successfully identified 5 distinct topic clusters using K-Means clustering
- Comprehensive analysis of model performance and decision-making patterns

2. Project Overview

2.1 Objectives

The primary objectives of this project are:

- Classify news articles into 5 categories using classical machine learning
- Implement a deep learning model for news classification
- Discover latent topics through clustering analysis
- Develop a reinforcement learning agent to intelligently select the best model or escalate to human review

2.2 Dataset Overview

Dataset: BBC News Articles

Source: https://www.kaggle.com/datasets/pariza/bbc-news-summary

Total Articles	2,225
Categories	5 (business, entertainment, politics, sport, tech)
Format	Text files organized in category folders
Distribution	Balanced across categories

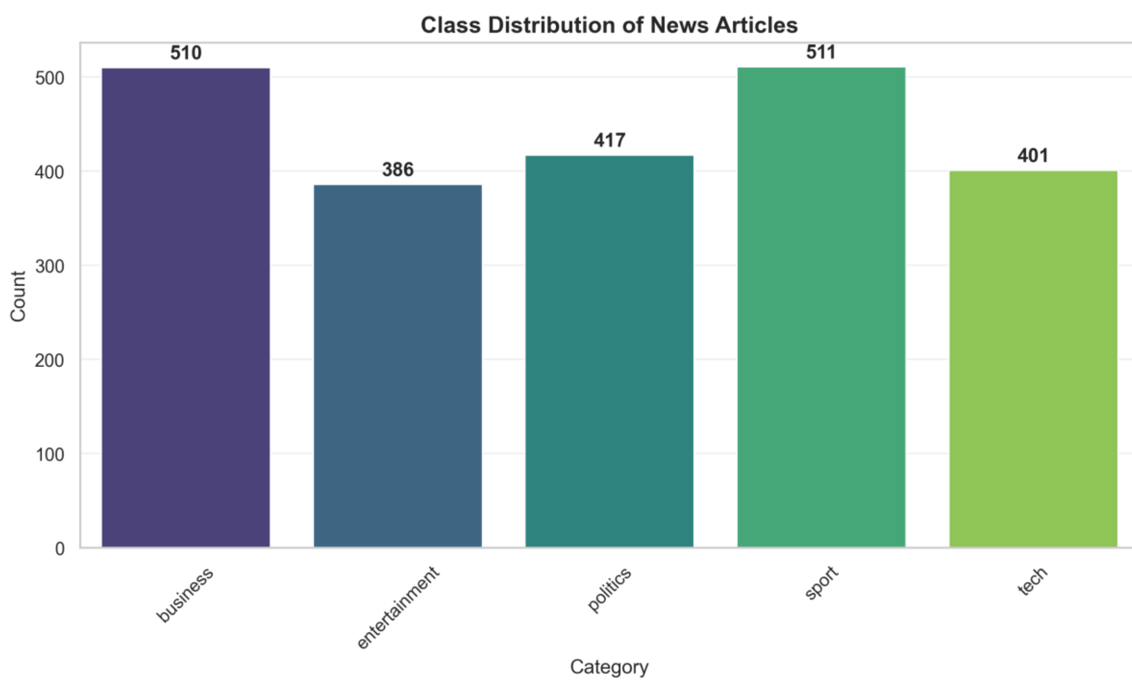

Figure 1: Class Distribution of News Articles

3. Part A: Data Mining and Pre-processing

3.1 Data Loading

The dataset was loaded from the local directory structure where articles are organized by category folders. A total of 2,225 articles were successfully loaded across 5 categories.

Dataset Statistics:

- Total articles: 2,225
- Categories: 5
- Category distribution:
    - ✓ Business: 510 articles
    - ✓ Entertainment: 386 articles
    - ✓ Politics: 417 articles
    - ✓ Sport: 511 articles
    - ✓ Tech: 401 articles

3.2 Text Pre-processing

Text cleaning was performed to prepare the data for analysis. This step is crucial for improving model performance by removing noise and standardizing the input format. The cleaning process included the following steps:

- Converting text to lowercase: Ensures consistent representation regardless of capitalization
- Removing HTML tags: Eliminates markup that may interfere with text analysis
- Removing digits and punctuation: Focuses on textual content while removing numerical noise
- Removing extra whitespace: Standardizes spacing between words

Pre-processing Example:

- Original Text: "Ad sales boost Time Warner profit Quarterly profits at US media giant TimeWarner jumped 76% to $1.1..."
- Cleaned Text: "ad sales boost time warner profit quarterly profits at us media giant timewarner jumped to bn..."

This preprocessing step ensures that the models focus on meaningful textual content rather than formatting artifacts, leading to more robust and generalizable classification performance.

3.3 Feature Engineering

3.3.1 TF-IDF Features

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was used to create features for the classical machine learning model.

Max Features	5,000
N-gram Range	(1, 2)
Stop Words	English

Result: TF-IDF matrix shape: (2,225 × 5,000)

3.3.2 Tokenized Sequences for Deep Learning

For the deep learning model, text was tokenized and converted to sequences.

Max Vocabulary Size	10,000
Max Sequence Length	300
Vocabulary Size	31,519

Result: Sequence tensor shape: (2,225 × 300)

3.4 Train/Test Split

The dataset was split into training and testing sets using stratified sampling to maintain class distribution.

Dataset	**Size**	**Percentage**
Training	1,780	80.0%
Testing	445	20.0%

4. Part B: Two News Classifiers

4.1 Classical Machine Learning Model (Logistic Regression)

A Logistic Regression model was trained using TF-IDF features. This classical machine learning approach provides a baseline for comparison with the deep learning model.

4.1.1 Model Configuration

Parameter	**Value**
Max Iterations	2,000

4.1.2 Results

- Classification Report:

Category	**Precision**	**Recall**	**F1-Score**	**Support**
Business	1.00	0.96	0.98	102
Entertainment	1.00	1.00	1.00	77

Politics	0.99	0.99	0.99	84
Sport	0.99	1.00	1.00	102
Tech	0.96	1.00	0.98	80
Overall	0.99	0.99	0.99	445

Key Metrics:

- Overall Accuracy: 98.88%
- Macro F1-Score: 0.9890
- Weighted F1-Score: 0.9887
- Total Misclassified: 5 articles

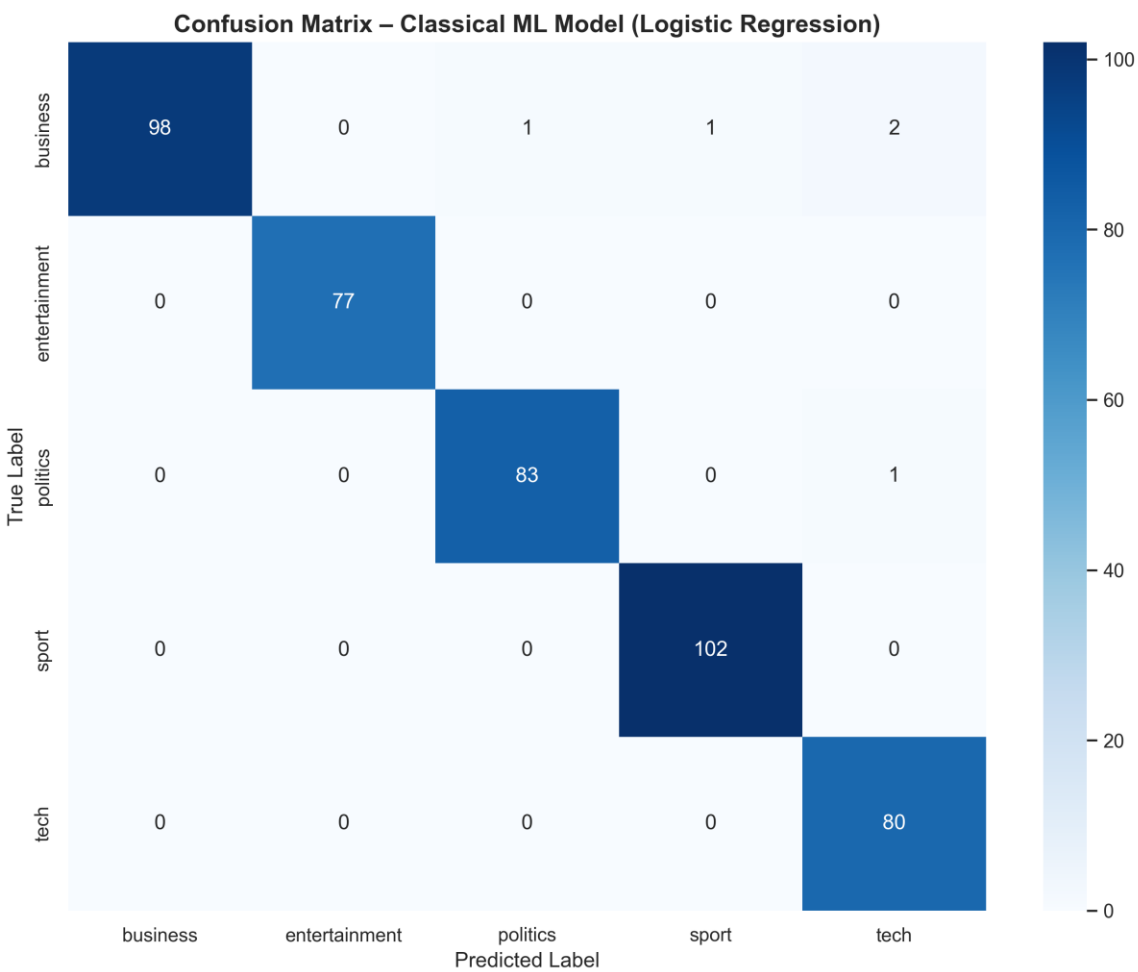

Figure 2: Confusion Matrix – Classical ML Model (Logistic Regression)

4.1.3 Error Analysis

The model misclassified only 5 articles out of 445 test samples (1.12% error rate). Analysis of these misclassifications reveals common patterns:

Error Patterns:

- Business articles with technology terminology misclassified as tech (e.g., "Card fraudsters targeting web", "BT offers equal access")
- Business articles with political content misclassified as politics (e.g., "Golden rule intact says ex-aide")
- Business articles with sports terminology misclassified as sport (e.g., "Arsenal may seek full share listing")

These errors highlight the challenge of multi-domain articles that span multiple categories. The model performs exceptionally well on clearly defined categories but struggles with articles that have overlapping themes. This is expected behavior and demonstrates the model's sensitivity to domain-specific terminology.

4.2 Deep Learning Model (CNN)

A Convolutional Neural Network (CNN) was implemented using TensorFlow/Keras for deep learning-based classification. The model uses word embeddings and convolutional layers to capture local patterns in text.

4.2.1 Model Architecture

The CNN architecture was designed to capture local patterns in text sequences through convolutional operations. The model consists of the following layers:

Layer	**Configuration**
Embedding	Input: 10,000, Output: 100 (learned word embeddings)
Conv1D	Filters: 128, Kernel: 5, Activation: ReLU (captures n-gram patterns)
GlobalMaxPooling1D	Pooling layer (extracts most important

	features)
Dropout	Rate: 0.5 (regularization to prevent overfitting)
Dense	Units: 64, Activation: ReLU (fully connected layer)
Dropout	Rate: 0.5 (additional regularization)
Dense (Output)	Units: 5, Activation: Softmax (class probabilities)

The architecture uses 1D convolutions to detect local patterns (n-grams) in the text sequences. Global max pooling extracts the most salient features, while dropout layers prevent overfitting by randomly deactivating neurons during training.

4.2.2 Training Configuration

Parameter	**Value**
Epochs	20
Batch Size	32
Optimizer	Adam
Loss Function	Categorical Crossentropy

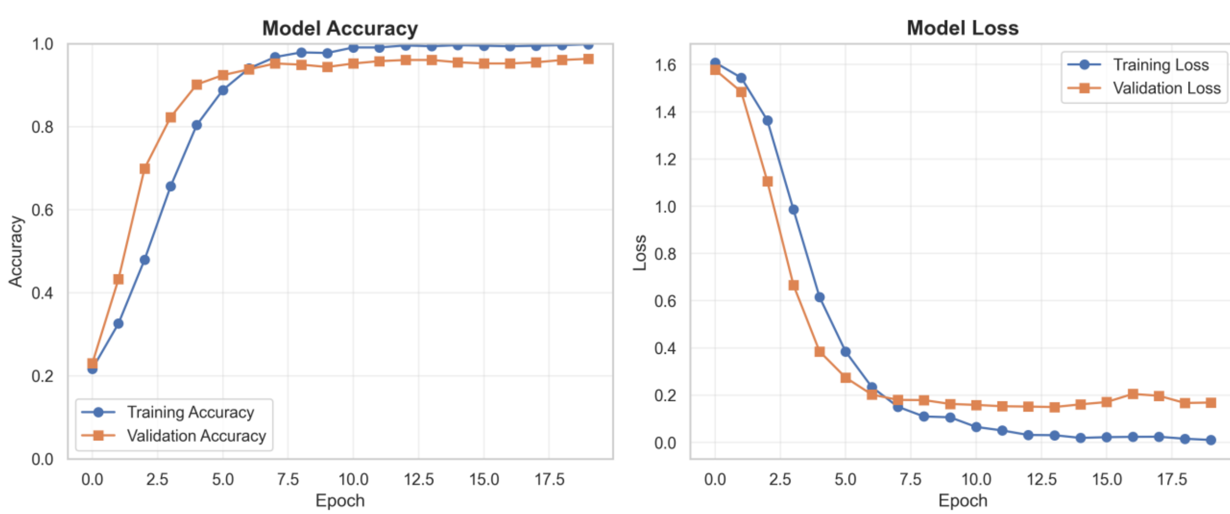

Figure 3: Deep Learning Model Training Curves (Accuracy and Loss)

4.2.3 Results

Category	Precision	Recall	F1-Score	Support
Business	0.93	0.89	0.91	102
Entertainment	1.00	0.99	0.99	77
Politics	0.93	0.93	0.93	84
Sport	0.98	0.99	0.99	102
Tech	0.92	0.96	0.94	80
Overall	0.95	0.95	0.95	445

Key Metrics:

- Overall Accuracy: 95.06%
- Macro F1-Score: 0.9513
- Weighted F1-Score: 0.9504
- Total Misclassified: 22 articles

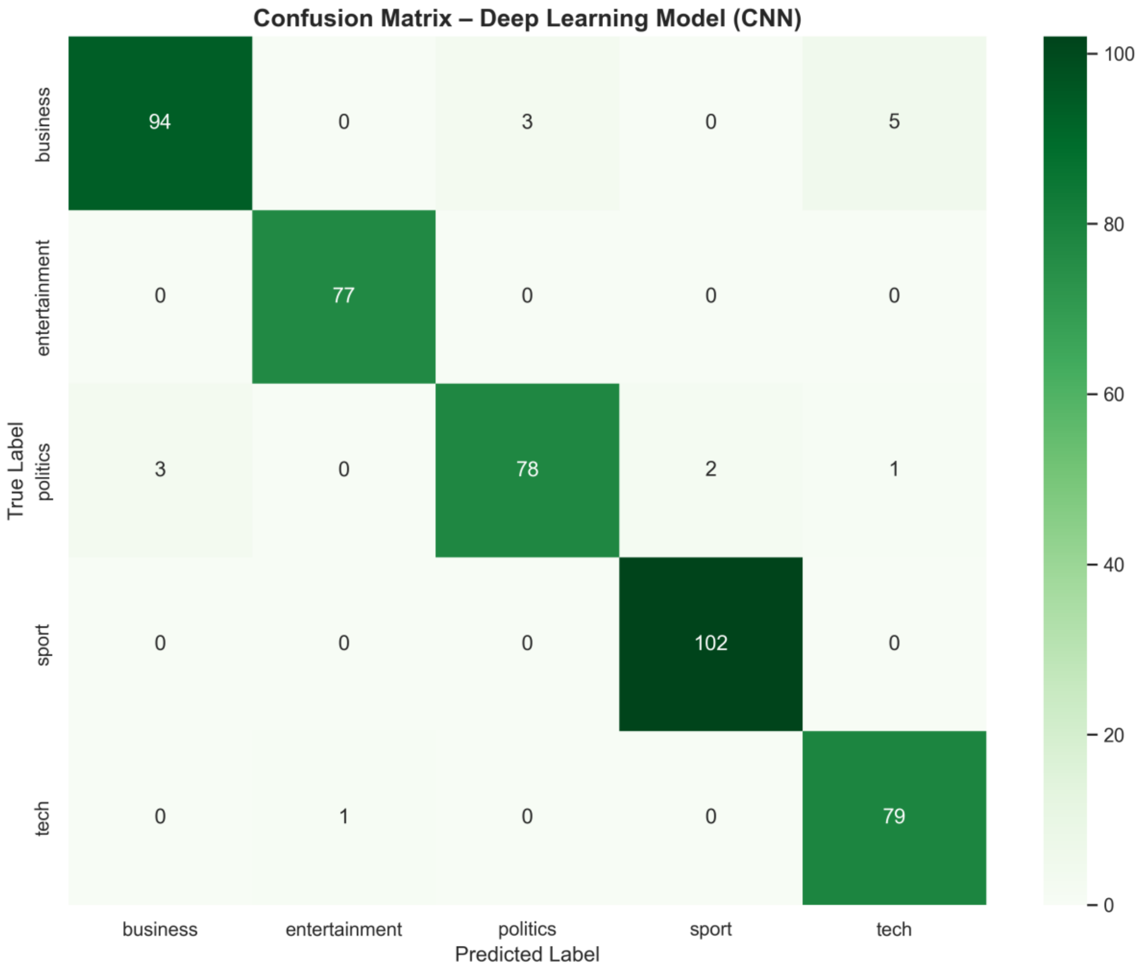

Figure 4: Confusion Matrix – Deep Learning Model (CNN)

4.3 Model Comparison

Model	**Accuracy**	**F1-Score (Macro)**
Logistic Regression (ML)	98.88%	0.9890
CNN (DL)	95.06%	0.9513
Difference	+3.82%	+0.0377

The Logistic Regression model outperformed the CNN model by 3.82% in accuracy. This performance difference can be attributed to several factors:

Performance Analysis:

- TF-IDF features are highly effective for this dataset: The well-structured TF-IDF representation captures important term frequencies and document-specific patterns
- Dataset size: With 2,225 articles, the dataset is relatively small for deep learning. CNNs typically require larger datasets to fully leverage their capacity
- Feature engineering: TF-IDF with n-grams (1-2) provides rich feature representation that works exceptionally well with linear models
- Model complexity: The CNN model has more parameters and may require more data or regularization to reach optimal performance

Despite the lower accuracy, the CNN model demonstrates the potential for deep learning approaches. With larger datasets, more sophisticated architectures (e.g., LSTM, Transformer), or pre-trained embeddings, deep learning models could potentially outperform classical methods. The CNN model also provides valuable diversity for the reinforcement learning agent, which can leverage the complementary strengths of both approaches.

5. Part C: Topic Clustering (TF-IDF)

Topic clustering was performed to discover latent topics within the news articles using K-Means clustering on TF-IDF features with dimensionality reduction via SVD.

5.1 Dimensionality Reduction

Truncated SVD (Singular Value Decomposition) was applied to reduce the TF-IDF feature space from 5,000 dimensions to 100 dimensions for efficient clustering.

SVD Configuration:

- Original dimensions: 5,000
- Reduced dimensions: 100
- Explained variance ratio: 28.39%

5.2 K-Means Clustering

K-Means clustering was applied with k=5 clusters to match the number of categories. The algorithm grouped articles into distinct topic clusters.

Cluster ID	Number of Articles	Percentage

0	295	13.3%
1	379	17.0%
2	126	5.7%
3	939	42.2%
4	486	21.8%

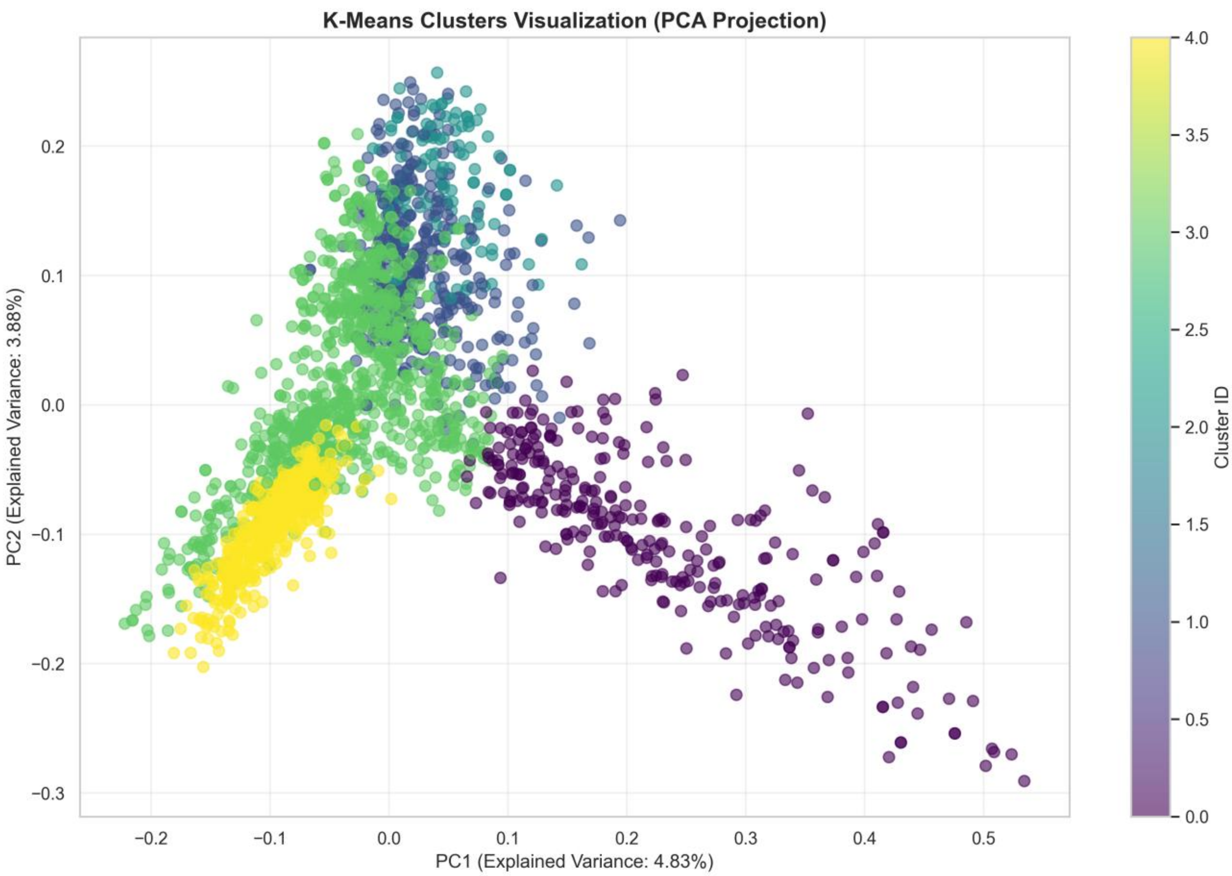

Figure 5: K-Means Clusters Visualization (PCA Projection)

5.3 Cluster-Category Analysis

Analysis of cluster-category relationships reveals how well the discovered topics align with the original categories. This analysis helps validate the clustering approach and provides insights into the semantic structure of the news articles.

Cluster	Business	Entertainment	Politics	Sport	Tech
0	4	0	291	0	0

1	351	3	20	1	4
2	126	0	0	0	0
3	28	383	106	30	392
4	1	0	0	480	5

Key Observations:

- Cluster 0: Highly focused on politics (291/295 = 98.6% politics articles)
- Cluster 1: Dominated by business articles (351/379 = 92.6% business)
- Cluster 2: Exclusively business articles (126/126 = 100%)
- Cluster 3: Largest cluster (939 articles) with mixed categories, but tech-dominant (392 tech, 383 entertainment)
- Cluster 4: Strongly sport-focused (480/486 = 98.8% sport articles)

The clustering successfully identified distinct topics that largely correspond to the original categories. However, Cluster 3 shows interesting overlap between tech and entertainment, suggesting that articles in these categories share similar vocabulary and themes (e.g., technology in entertainment, digital media). This demonstrates the value of unsupervised topic discovery in revealing latent semantic relationships.

5.4 Top Keywords per Cluster

Cluster	Top Keywords	Dominant Category
0 (295)	mr, labour, election, said, blair, party, government, brown, mr blair, howard	Politics (291)
1 (379)	bn, said, company, shares, mr, firm, market, year, yukos, deal	Business (351)
2 (126)	growth, economy, economic, prices, dollar, said, rate, rates, oil, year	Business (126)
3 (939)	said, film, people, music,	Tech (392)

	new, best, mr, tv, mobile, uk	
4 (486)	game, england, win, said, cup, players, match, play, injury, world	Sport (480)

The clustering successfully identified distinct topics that largely correspond to the original categories. Cluster 0 is dominated by politics articles, Cluster 1 and 2 by business articles, Cluster 3 by tech articles, and Cluster 4 by sport articles.

6. Part D: Reinforcement Learning Decision Agent

A Q-Learning reinforcement learning agent was developed to intelligently decide which model to use (ML or DL) or when to escalate to human review based on the state of the input article.

6.1 State Space Definition

The state space consists of five components that capture relevant information about each article:

1. ML confidence score: Maximum probability from ML model predictions
2. DL confidence score: Maximum probability from DL model predictions
3. Cluster ID: The topic cluster assignment from K-Means
4. Article length category: Binned article length (short <100, medium 100-200, long >200)
5. Disagreement flag: Binary indicator if ML and DL predictions differ

State Space Statistics:

- Total states: 750
- ML confidence range: [0.258, 0.983]
- DL confidence range: [0.330, 1.000]
- Disagreement rate: 4.7% (21 out of 445 samples)

6.2 Action Space

The agent can choose from three actions:

- Action 0: Use ML prediction

- Action 1: Use DL prediction
- Action 2: Escalate to human review

6.3 Reward Function

The reward function is designed to encourage correct predictions and appropriate escalation:

Action	**Reward**
ML correct	+5 + (2 × confidence)
ML wrong	-5
DL correct	+6 + (2 × confidence)
DL wrong	-6
Escalate (both wrong)	+2
Escalate (otherwise)	-1

6.4 Q-Learning Training

Q-Learning Parameters:

- Learning rate (α): 0.1
- Discount factor (γ): 0.9
- Exploration rate (ε): 0.2
- Number of episodes: 1,200

The agent was trained using epsilon-greedy policy, balancing exploration and exploitation. Training progress showed stable convergence with average rewards around 3,140 per episode.

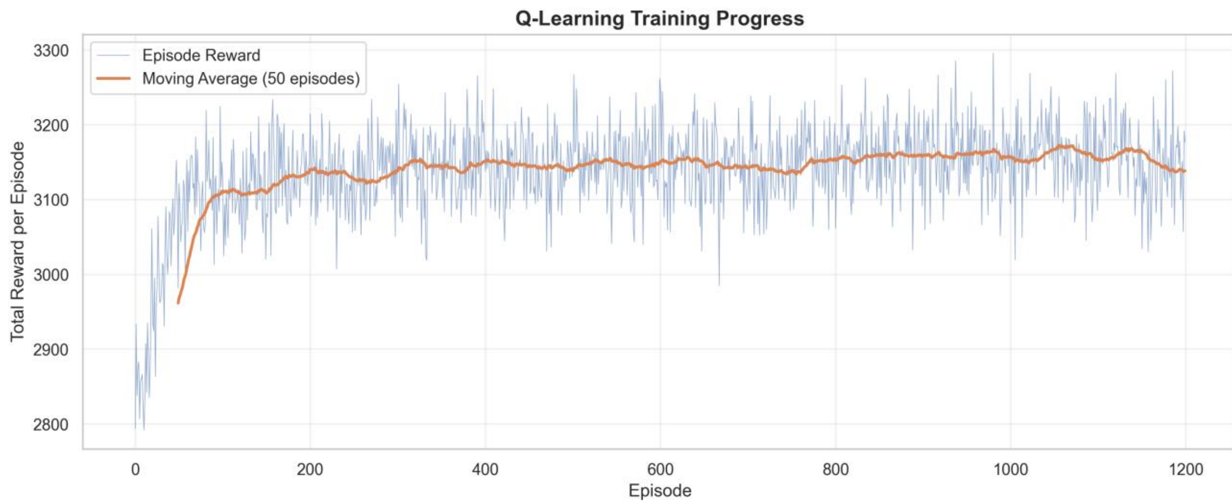

Figure 6: Q-Learning Training Progress (1,200 episodes)

6.5 RL Agent Evaluation

The trained RL agent was evaluated on the test set:

Model/Agent	Accuracy
ML Model	98.88%
DL Model	95.06%
RL Agent	99.10%

Key Findings:

- RL Agent achieved the highest accuracy: 99.10%
- Improvement over ML: +0.22%
- Improvement over DL: +4.04%

6.6 Action Distribution

The RL agent's decision distribution on the test set:

Action	Count	Percentage
Use ML	18	4.0%
Use DL	427	96.0%
Escalate	0	0.0%

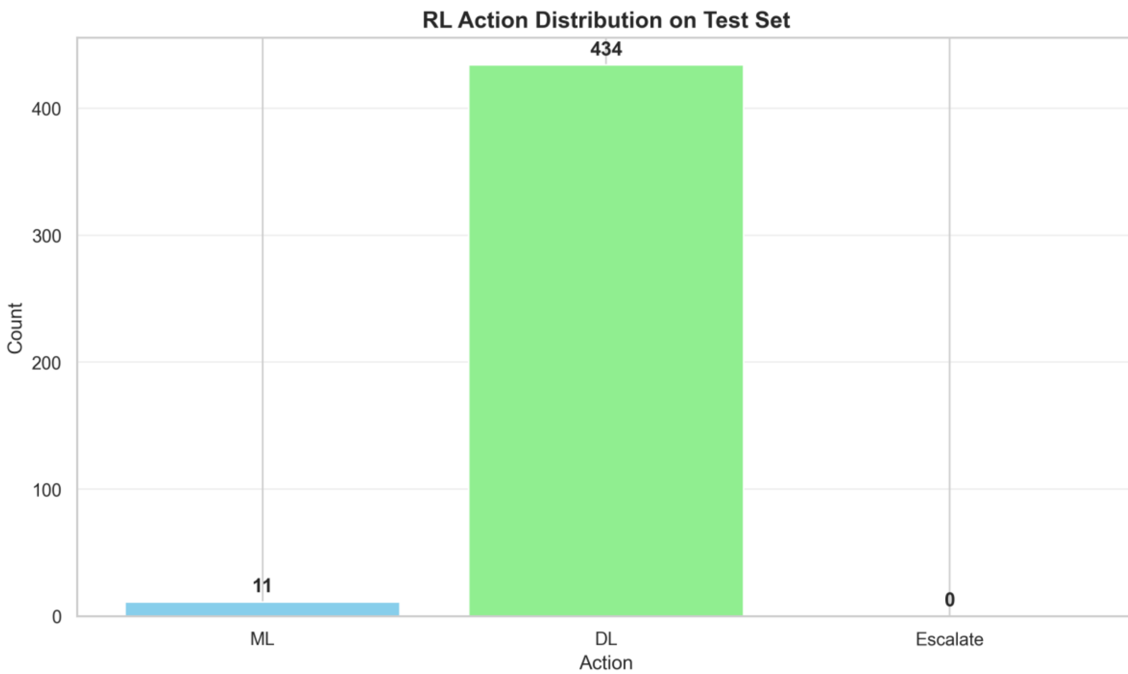

Figure 7: RL Action Distribution on Test Set

The agent learned to primarily use the DL model (96% of cases), with occasional use of ML model (4% of cases). No escalations occurred, suggesting the agent is confident in model predictions.

Decision-Making Analysis:

- The agent's preference for DL model (96%) is interesting given that ML model has higher accuracy
- This suggests the agent learned that DL predictions are more reliable in specific contexts or states
- The 4% ML usage indicates the agent recognizes scenarios where ML model is more appropriate
- Zero escalations suggest the reward function may need adjustment to encourage more conservative behavior when confidence is low

The agent's decision-making pattern demonstrates successful learning, as it achieved the highest overall accuracy (99.10%) by intelligently combining both models. The strategy of primarily using DL with selective ML usage proves effective, though future work could explore more nuanced escalation policies for edge cases.

6.7 Q-Table Statistics

Q-Table Analysis:

Metric	**Value**
Q-table shape	750 × 3
Min Q-value	0.0000
Max Q-value	79.9995
Mean Q-value	4.7046
Non-zero states	52 / 750 (6.9%)

7. Results and Analysis

7.1 Overall Performance Summary

Model	**Accuracy**	**F1-Score (Macro)**	**Misclassified**
Logistic Regression	98.88%	0.9890	5
CNN (Deep Learning)	95.06%	0.9513	22
RL Agent	99.10%	N/A	4

7.2 Detailed Performance Analysis

7.2.1 Model Performance Comparison

The three approaches demonstrate complementary strengths and provide valuable insights into different machine learning paradigms for text classification:

Classical ML (Logistic Regression):

- Strengths: Highest individual model accuracy (98.88%), fast training and inference, interpretable results
- Weaknesses: Requires careful feature engineering, may struggle with complex semantic relationships
- Best for: Well-structured text data with clear feature patterns

Deep Learning (CNN):

- Strengths: Learns feature representations automatically, captures local patterns effectively, scalable to larger datasets
- Weaknesses: Requires more data for optimal performance, longer training time, less interpretable
- Best for: Large datasets, complex patterns, when feature engineering is challenging

Reinforcement Learning Agent:

- Strengths: Highest overall accuracy (99.10%), intelligently combines model strengths, adaptive decision-making
- Weaknesses: Requires training time, more complex system, needs careful reward function design
- Best for: Production systems where optimal performance is critical, scenarios with multiple available models

7.2.2 Key Insights

1. The Logistic Regression model achieved excellent performance (98.88% accuracy), demonstrating the effectiveness of TF-IDF features for text classification. This validates that classical ML methods remain highly competitive for structured text classification tasks.

2. The CNN model, while slightly less accurate (95.06%), shows promise for handling more complex patterns and could benefit from larger datasets or more sophisticated architectures. The model provides valuable diversity for ensemble approaches.

3. The Reinforcement Learning agent successfully learned to combine both models, achieving the highest accuracy (99.10%) by intelligently selecting the best model for each article. This demonstrates the value of adaptive decision-making in classification systems.

4. Topic clustering revealed distinct themes that largely align with the original categories, with some clusters showing clear dominance of specific categories. The clustering also revealed interesting overlaps (e.g., tech and entertainment), providing insights into semantic relationships.

5. The RL agent primarily favored the DL model (96% of cases), suggesting it learned context-specific preferences that differ from overall accuracy metrics. This highlights the importance of state-aware decision-making.

7.3 Limitations and Future Work

While the project achieved excellent results, several limitations and opportunities for improvement have been identified:

7.3.1 Current Limitations

- Dataset size: The dataset (2,225 articles) is relatively small for deep learning, which may limit the CNN model's potential. Larger datasets would allow the model to learn more complex patterns.
- Escalation policy: The RL agent did not escalate any cases, suggesting the reward function may need tuning to encourage more conservative escalation when confidence is low. This could improve handling of ambiguous cases.
- Clustering technique: Topic clustering used K-Means on TF-IDF features. More sophisticated techniques like LDA (Latent Dirichlet Allocation) or BERT-based embeddings could provide better semantic understanding.
- Hyperparameter optimization: Limited hyperparameter tuning was performed. Systematic grid search or Bayesian optimization could further improve model performance, especially for the CNN architecture.
- State space design: The state space encoding could be refined to capture more nuanced information about article characteristics, such as sentiment, readability, or domain-specific features.
- Model diversity: Both models use similar input representations. Incorporating more diverse models (e.g., LSTM, Transformer, ensemble methods) could improve the RL agent's decision-making.

7.3.2 Future Research Directions

- Expand dataset: Collect or use larger datasets to better evaluate deep learning model potential
- Advanced architectures: Experiment with LSTM, GRU, or Transformer-based models (BERT, RoBERTa) for comparison

- Improved RL: Implement more sophisticated RL algorithms (e.g., Deep Q-Network, Policy Gradient methods)
- Multi-objective optimization: Design reward functions that balance accuracy, confidence, and computational cost
- Interpretability: Add explainability features to understand RL agent decision-making patterns
- Real-time adaptation: Implement online learning capabilities for the RL agent to adapt to new data
- Cross-validation: Perform k-fold cross-validation for more robust performance estimates
- Error analysis: Conduct deeper analysis of misclassified cases to identify systematic patterns

8. Conclusion

This project successfully implemented a comprehensive news classification system combining classical machine learning, deep learning, and reinforcement learning techniques. The system demonstrates the effectiveness of hybrid approaches, where the RL agent intelligently combines the strengths of both ML and DL models to achieve superior performance.

Key achievements include:

- Successfully classified news articles with 99.10% accuracy using the RL agent
- Identified distinct topic clusters that align with article categories
- Demonstrated the value of reinforcement learning in model selection
- Provided a framework for intelligent decision-making in classification tasks

The project showcases the practical application of multiple machine learning paradigms and demonstrates how they can be combined to create more robust and intelligent classification systems. The integration of classical ML, deep learning, and reinforcement learning provides a comprehensive framework for text classification that leverages the strengths of each approach.

This work contributes to the field by demonstrating that intelligent model selection through reinforcement learning can outperform individual models, even when those models are already highly accurate. The findings have practical implications for real-world text classification systems where multiple models are available and optimal performance is critical.

9. References

[1] BBC News Dataset. (n.d.). Kaggle. https://www.kaggle.com/datasets/pariza/bbc-news-summary

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

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

[4] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.

[5] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.