BBC Text Classification Report

1. Introduction:

Natural Language Processing (NLP) is vital in AI for understanding and generating human language. Text classification, a core NLP task, is applied in areas like sentiment analysis, spam detection, and customer feedback. This project focuses on accurately categorizing news articles into topics such as business, politics, sports, and entertainment. Effective text classification addresses challenges in managing unstructured text data, improving applications like recommendation systems, news aggregators, and content filters.

2. Scope and Importance:

This report addresses the problem of accurate text classification by applying both traditional machine learning techniques (e.g., Logistic Regression, Support Vector Machine) and modern deep learning methods (e.g., Long Short-Term Memory networks, Gated Recurrent Unit). By exploring these methodologies, it provides insights into their comparative effectiveness for categorizing news articles into topics like business, politics, sports, and entertainment. Topic classification presents nuanced challenges due to overlapping vocabulary and complex language constructs across topics. The project critically evaluates how different models handle these issues, contributing to the advancement of NLP.

Text classification has become pivotal in managing unstructured text in the digital age. The global NLP market is projected to reach $61.03 billion by 2028, growing at a CAGR of 25.7% from 2021 to 2028, according to Fortune Business Insights (Anon., 2022) (Anon., 2022). This growth underscores the reliance on NLP for automating tasks like content moderation, personalized recommendations, and information retrieval. Despite advancements, challenges remain in disambiguating closely related categories (e.g., "business" vs. "politics"). This project bridges gaps by testing traditional and deep learning models for nuanced classification tasks.

3. Background Review:

Recent advancements in natural language processing (NLP), particularly in text classification, have spanned traditional statistical methods to sophisticated deep learning architectures, focusing on improving performance, scalability, and interpretability. Text classification has evolved significantly, with notable contributions addressing challenges like model complexity, computational costs, and domain-specific applications.

A study provides a unified framework for understanding deep NLP models, emphasizing model interpretability (Zhen Li, 2022). It introduces multi-level visualization techniques to uncover word contributions and relationships across model layers, aiding developers in understanding predictions. This framework also highlights challenges like mislabeled data and uneven distributions, enabling more effective debugging and refinement of architectures such as RNNs, CNNs, and Transformers. The framework is particularly relevant for large-scale tasks requiring detailed analysis of model behavior.

(Amrita Bhattacharjee, 2024) explores a novel approach to causal explainability for black-box text classifiers using large language models (LLMs). The proposed three-step pipeline uses LLMs to extract latent features, associate them with input features, and generate counterfactual explanations by minimally altering input text. Results highlight the pipeline's effectiveness, with GPT-4 outperforming other models in accuracy, semantic similarity, and counterfactual quality. This methodology advances explainability in NLP by leveraging LLMs' contextual understanding, bridging gaps in causal inference, and enhancing interpretability for real-world applications in text classification.

(Shervin Minaee, 2021) provides a comprehensive review of over 150 deep learning models developed for text classification tasks such as sentiment analysis, news categorization, and natural language inference. It highlights the superiority of deep learning over traditional machine learning methods, showcasing advances in architectures like RNNs, CNNs, attention mechanisms, and Transformers. Quantitative analysis reveals that models like BERT and GPT consistently outperform earlier approaches on benchmark datasets, demonstrating their effectiveness in extracting contextual and semantic features. Additionally, the study discusses the challenges of scalability, interpretability, and domain adaptation, suggesting future research directions to address these limitations.

(Christopher Schröder, 2024) explores self-training to enhance active learning for text classification, introducing HAST, a novel strategy that uses pseudo-labels to minimize labeled data dependency. Evaluated across four benchmarks, HAST matches state-of-the-art results using just 25% of the data, highlighting its efficiency in integrating certainty-based self-training with active learning to achieve robust text classification performance.

(Zhiqiang Wang, 2024) introduces a streamlined text classification framework leveraging Large Language Models (LLMs), emphasizing minimal preprocessing and user expertise. It demonstrates LLMs outperform traditional models in tasks like sentiment analysis and spam detection across diverse datasets. Fine-tuning significantly enhances accuracy and reliability, though challenges like inconsistent outputs and computational demands persist.

4. SMART Objectives:

The SMART objectives for the BBC text classification project are refined to align with technical precision, measurable benchmarks, evidence-based feasibility, meaningful contribution, and adherence to a specific timeline.

4.1. Specific

The project focuses on building and evaluating text classification models using the BBC News dataset. The technical scope includes the implementation of two traditional machine learning models, such as Logistic Regression and Support Vector Machines (SVM), and two deep learning models, including LSTM and GRUs, for text classification. Specific steps include data preprocessing (e.g., tokenization, stop-word removal etc.), training the models, and comparing their performance.

4.2. Measurable

The project's success is measured by achieving a classification accuracy of at least 85%, based on benchmarks established in prior studies on similar datasets. For example, previous work using traditional classifiers on news datasets has achieved accuracies around 80-85%, while modern deep learning models have exceeded 90% in many cases (Liu et al., 2022; Zhang et al., 2024). The evaluation will also include visualizations such as confusion matrices and ROC curves for clarity.

4.3. Achievable

The proposed text classification objectives are feasible, leveraging proven techniques like Naive Bayes, SVM, and deep learning models (e.g., LSTMs, Transformers). Pre-trained embeddings like BERT and GloVe enhance performance while optimizing computational efficiency, as demonstrated in similar studies on structured and unstructured text data (Amrita Bhattacharjee, 2024), but are not used.

4.4. Relevant

This project advances NLP by comparing traditional and deep learning models for text classification, emphasizing accuracy, interpretability, and efficiency. Addressing applications like news categorization, spam detection, and sentiment analysis, it tackles a relevant and impactful problem in the field.

4.5. Time-Bound

The project follows a structured timeline, with tasks progressing weekly from data preprocessing to model evaluation and report compilation. The coding took 6 days, while the report required one week to structure.

5. Dataset:

The BBC News dataset, sourced from **Kaggle**, is a publicly available textual dataset. It has been widely utilized in natural language processing (NLP) research due to its simplicity and, balance.

5.1. Properties of the Dataset

- **Source, Size, and Shape**: The dataset comprises 2,225 labeled news articles originally sourced from BBC News and hosted on Kaggle. Each article is categorized into one of five topics: business, entertainment, politics, sports, or technology. The articles vary in length, ranging from concise summaries to comprehensive reports.
- **Class Balance**: The dataset is balanced, with an approximately equal number of samples in each category. This ensures fair representation of all classes during training, reducing biases and enabling unbiased model evaluation.
- **Structured and Usable**: Although the text content is unstructured, the dataset itself is organized and labeled. Proper preprocessing techniques such as tokenization, vectorization, and stop-word removal are necessary to convert raw text into formats suitable for machine learning models.

5.2. Suitability for the Problem

Strengths

1. **Relevance to Real-World Applications**: The dataset captures a typical task in NLP—news categorization—making it a realistic benchmark for evaluating model performance.
2. **Balanced Classes**: The even distribution of samples across categories ensures that models are not skewed toward specific labels, allowing for fair comparisons.
3. **Accessible and Reproducible**: Being available on Kaggle, the dataset is easy to access and widely used in research and education, ensuring reproducibility of experiments and findings.

Weaknesses

1. **Small Dataset Size**: With only 2,225 articles, the dataset may not fully support the capabilities of advanced deep learning models like LSTMs or Transformers, which typically perform better with larger datasets.
2. **Domain Specificity**: The dataset exclusively features BBC News articles, which may limit the generalizability of models trained on it to other news sources or domains with differing styles.
3. **Preprocessing Complexity**: The unstructured nature of the text requires preprocessing steps, such as tokenization, stop-word removal, and vectorization, adding complexity and potential sources of error to the machine learning pipeline.

6. Exploratory Data Analysis:

EDA is conducted by visualizing the class distribution, analyzing text length statistics, and generating WordClouds for different categories. Preprocessing includes:

- Removal of stopwords using NLTK.
- Tokenization and conversion of text into vectors using TF-IDF.

6.1. Class Distribution:

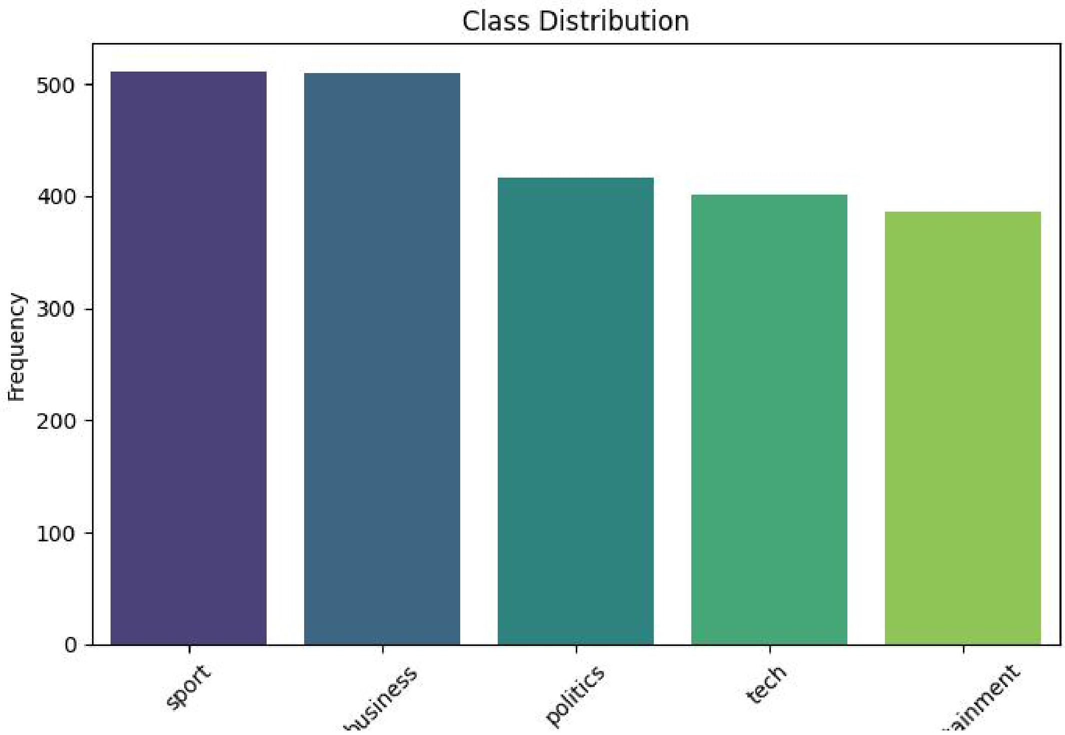


Figure 1: Class Distribution

Each bar in the class distribution represents the frequency of articles in each category, showing a relatively balanced dataset. Categories such as *sport* and *business* have slightly more samples compared to others, while *entertainment* has the fewest. This balance ensures that no class dominates the dataset.

6.2. Text Length Distribution:

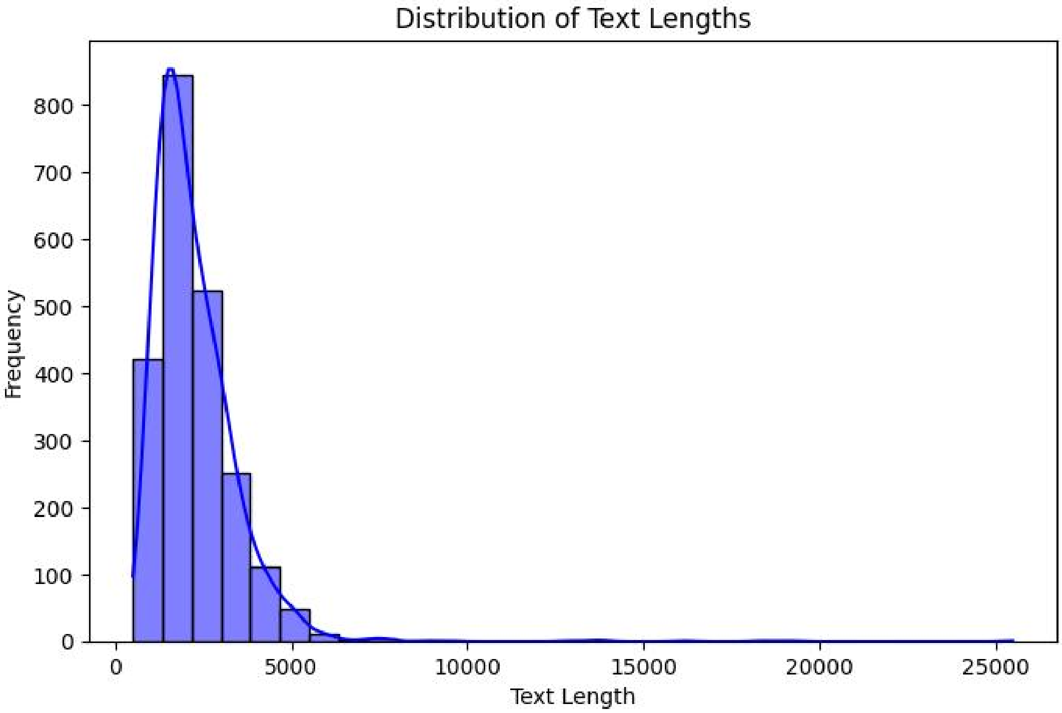


Figure 2: Distribution of Text Lengths

The histogram is right-skewed, indicating that the majority of articles are relatively short, with only a few significantly longer articles. The distribution peaks around a specific text length range, showing that most articles cluster within a certain length, but the long tail represents the outliers.

6.3. Variation in Text Length by category

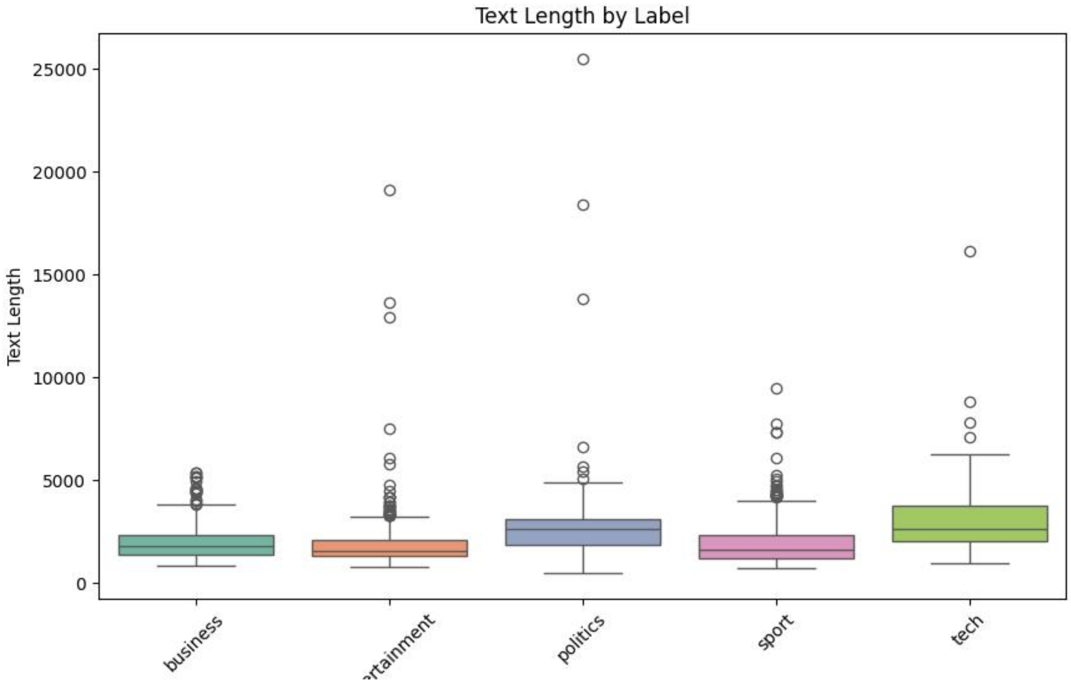
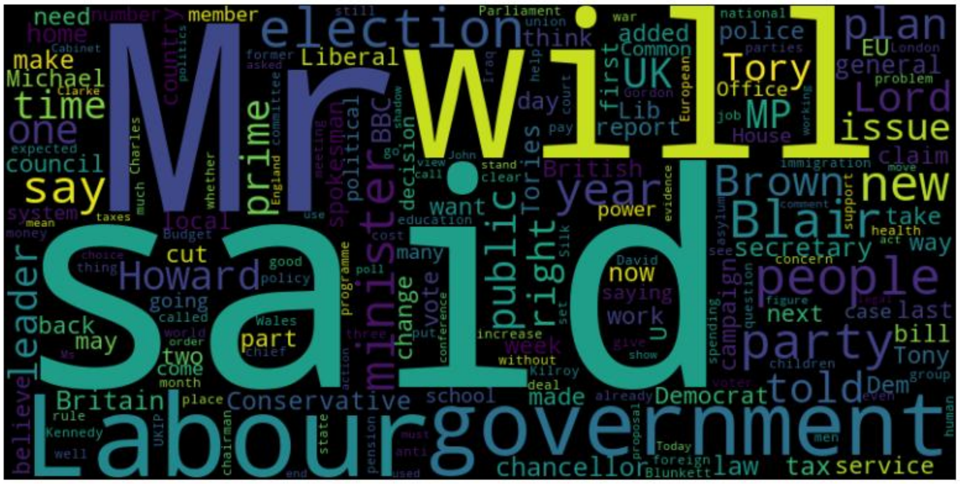


Figure 3: Text Length by Label

The central box for each label represents the interquartile range (IQR), while the whiskers indicate the overall range, excluding outliers. Categories like *tech* and *business* have larger median text lengths, while *entertainment* articles tend to be shorter. The presence of outliers, particularly in categories like *politics* and *business*, indicates some unusually long articles.

6.4. WordCloud for each category:

6.4.1. Politics:



WordCloud for Politics News

Figure 4: WordCloud for Politics

The word cloud for politics news prominently features terms such as "Mr," "Labour," "government," "said," and "will.". Words like "election," "Prime," "Minister," "Tory," and "Labour" indicate the significance of political discourse around elections, parliamentary actions, and party politics. This word cloud reflects the language of debates, public policies, and leadership discussions.

6.4.2. Sports:

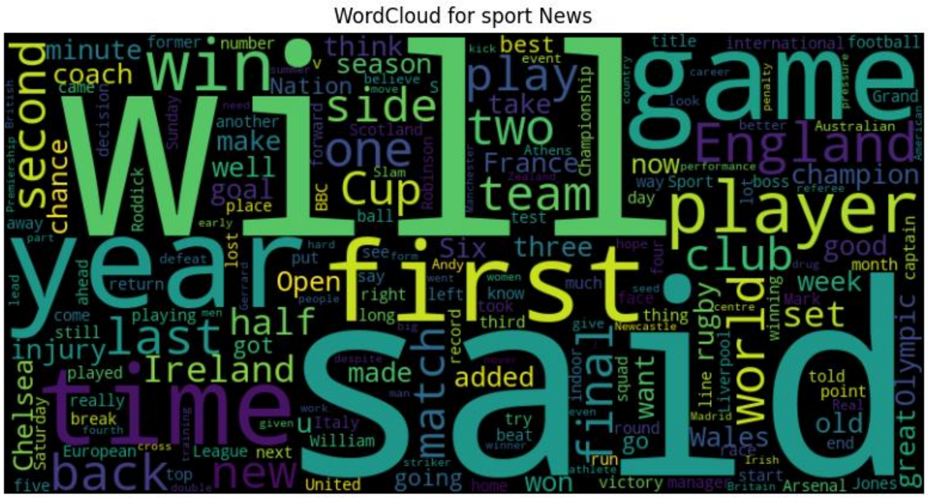


Figure 5: WordCloud for Sport News

In the sports news word cloud, words like "game," "win," "player," "team," and "year" dominate. This reflects discussions around matches, individual and team achievements, and timelines of significant sports events. Words such as "goal," "Cup," "England," and "club" show a focus on competitive games, national and international tournaments, and club-level sports activities.

6.4.3. Business:

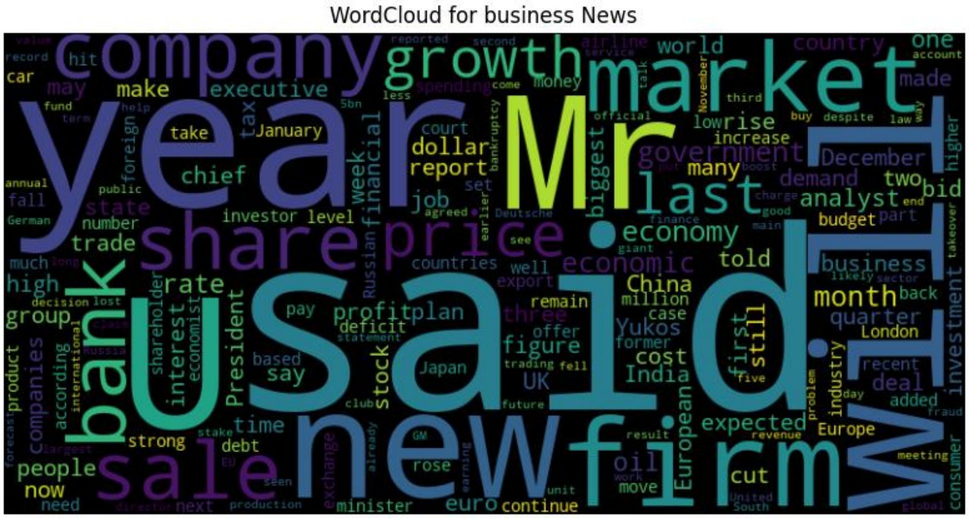


Figure 6: WordCloud for Business News

The business word cloud highlights terms like "year," "market," "firm," "Mr," and "growth." These indicate topics such as market performance, company strategies, and economic growth. Words like "share," "price," "bank," and "sale" point to financial transactions, investments, and market dynamics. This word cloud demonstrates the coverage of financial updates, corporate developments, and global economic trends.

6.4.4. Technology:

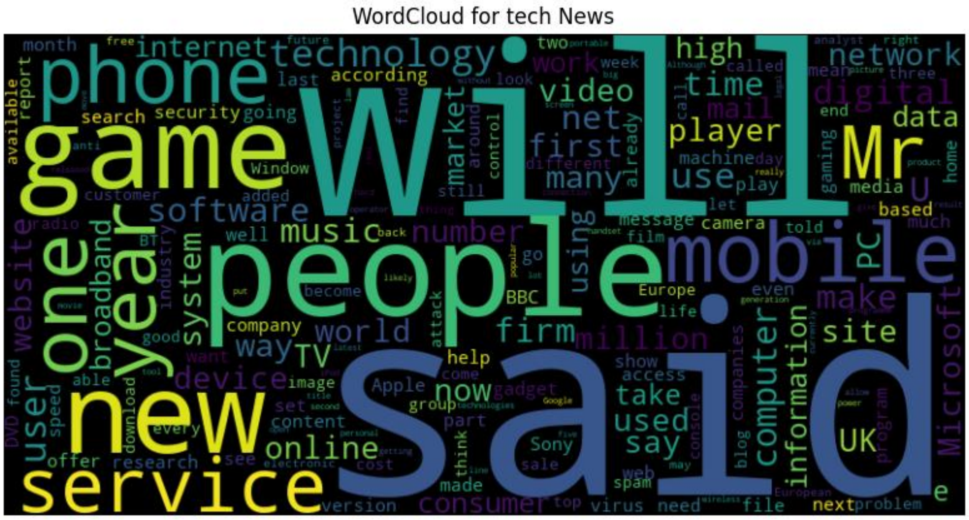


Figure 7: WordCloud for Tech News

In the tech news word cloud, terms like "said," "people," "new," "mobile," "phone," and "game" are prominent. These indicate a focus on technological advancements, consumer gadgets, and gaming. Words such as "service," "device," "network," and "technology" emphasize innovation, digital infrastructure, and user experiences.

6.4.5. Entertainment:

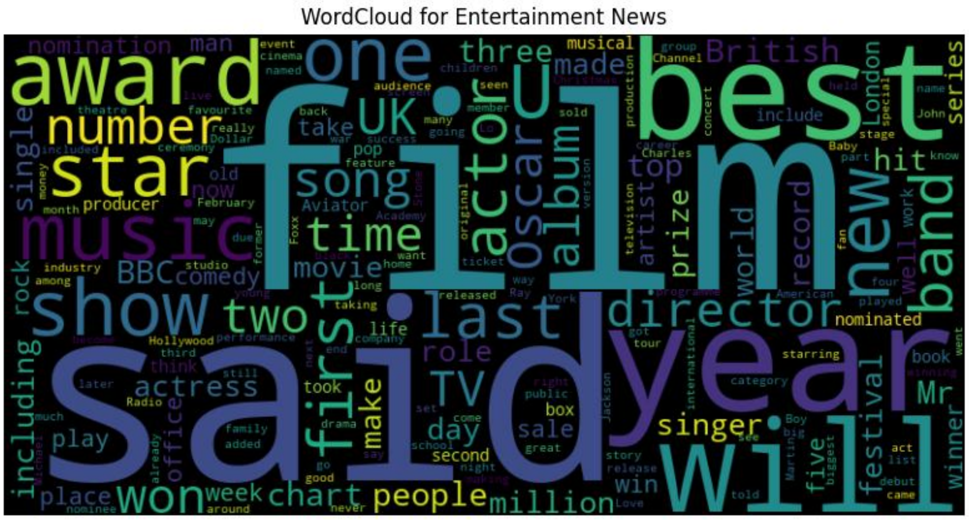

WordCloud for Entertainment News

Figure 8: WordCloud for Entertainment News

The entertainment news word cloud, featuring terms like "film," "director," "year," "best," "award," and "star," reflects a focus on the world of cinema, accolades, and prominent figures in the industry. Words such as "award" and "best" suggest discussions surrounding recognition, such as award ceremonies like the Oscars or Golden Globes. Terms like "director" and "film" highlight coverage of filmmaking and notable releases, while "star" points to the influence of actors and celebrities.

7. Comparison of Traditional Machine Learning Methods

7.1.1. Logistic Regression

Logistic Regression is a straightforward and interpretable algorithm commonly used for binary and multiclass classification. It performs well when features are linearly separable, especially when paired with text feature extraction techniques like TF-IDF. Its simplicity ensures computational efficiency. However, it struggles to model non-linear relationships and may underperform on datasets with complex patterns, which can limit its accuracy for intricate text classification tasks.

7.1.2. Support Vector Machine (SVM)

SVM constructs a hyperplane that maximizes the margin between classes, making it highly effective for high-dimensional data like TF-IDF feature representations of text. On balanced datasets like BBC News, SVM delivers robust performance. However, it is computationally intensive for large datasets and may falter in the presence of noisy or overlapping data.

7.1.3. Naive Bayes

Naive Bayes, a probabilistic algorithm assuming feature independence, is computationally efficient and effective for text classification. While its equal feature importance assumption limits performance on overlapping datasets, its simplicity and baseline reliability with TF-IDF features are advantageous.

7.1.4. K-Means Clustering

K-Means is an unsupervised algorithm that groups data into clusters based on feature similarity. While not directly suited for supervised tasks like classification, it can be useful for exploratory data analysis, offering insights into natural groupings within the dataset. However, it is sensitive to the initial placement of centroids and struggles with imbalanced data, reducing its effectiveness for tasks that require explicit label predictions.

7.1.5. Gradient Boosting Classifier (GBC)

Gradient Boosting is a powerful ensemble method that combines weak learners to build a strong classifier. However, it is computationally demanding, prone to overfitting, and requires extensive hyperparameter tuning. While GBC provides high predictive power, its complexity and focus on tabular data make it less practical for text classification tasks compared to simpler models like SVM and Naive Bayes.

7.1.6. Selected Models

Logistic Regression and SVM were chosen for this project due to their complementary strengths. **Logistic Regression** offers simplicity and computational efficiency, making it an excellent baseline model for text classification. **SVM**, on the other hand, excels in handling high-dimensional text data and delivers robust performance with linearly separable classes.

8. Comparison of Deep Learning Methods

8.1.1. Long Short-Term Memory (LSTM)

LSTMs, a type of RNN, capture long-term dependencies in sequential data, excelling in text classification by analyzing context in lengthy sequences. Despite challenges like computational intensity and overfitting on small datasets, their contextual capabilities make them highly effective.

8.1.2. Gated Recurrent Unit (GRU)

GRU is a simplified version of LSTM, with fewer parameters and gates, leading to faster training and reduced computational cost. GRUs are particularly advantageous for resource-constrained scenarios, as they require fewer computational resources. For the BBC dataset, GRU provides an efficient yet effective solution for text classification while preserving sequential information.

8.1.3. Convolutional Neural Networks (CNNs)

While CNNs are traditionally associated with image processing, they are also effective for text classification tasks by capturing local patterns, such as word n-grams. They are computationally efficient and well-suited for short text sequences. However, their inability to capture long-term dependencies in sequential data makes them less ideal for datasets like BBC News.

8.1.4. Bidirectional LSTM (BiLSTM)

BiLSTMs extend the standard LSTM by processing input sequences in both forward and backward directions. While BiLSTMs improve accuracy in tasks requiring deeper contextual understanding, they are computationally more expensive than regular LSTMs. This increased complexity makes them less practical for smaller datasets.

8.1.5. Transformers

Transformers, such as BERT, use attention mechanisms instead of recurrence to capture context and have set new benchmarks for text classification tasks. Their ability to understand context at a global level makes them highly effective for long and complex text. However, transformers are computationally intensive and require large-scale datasets to perform effectively, making them prone to overfitting on smaller datasets like BBC News.

8.1.6. Selected Methods: LSTM and GRU

LSTM and GRU are chosen for this project due to their ability to handle sequential data and contextual dependencies effectively. LSTM's strength lies in its capacity to capture long-term dependencies, which is valuable for understanding extended news articles. GRU complements this by offering a more computationally efficient approach, making it suitable for scenarios with limited resources.

9. Implementation and Refinement

The project leveraged Python and libraries like Pandas, NumPy, and NLTK for preprocessing, and Scikit-learn for TF-IDF vectorization and machine learning models (Logistic Regression, SVM). TensorFlow/Keras supported deep learning models (LSTM, GRU), while Matplotlib and Seaborn enabled data visualization for analysis and performance evaluation.

9.1. Procedures to Build, Train, and Test Pipelines

1. **Preprocessing**: Text data was cleaned by removing special characters, stopwords, and punctuation, followed by tokenization and TF-IDF vectorization for machine learning models.
2. **Building Models**: Machine learning pipelines included Logistic Regression and SVM. For deep learning, sequential architectures of LSTM and GRU were implemented.
3. **Training**: Models were trained on the preprocessed dataset using 80% of the data as training data and 20% for testing.
4. **Testing and Evaluation**: Metrics such as accuracy, F1-score were computed to evaluate the models. Confusion matrices were also plotted for visualizing performance.

9.2. Strategies for Fine-Tuning

9.2.1. Hyperparameter Tuning

Hyperparameter tuning was implemented to optimize the models. For machine learning pipelines, the following hyperparameters were tuned using Scikit-learn's GridSearchCV:

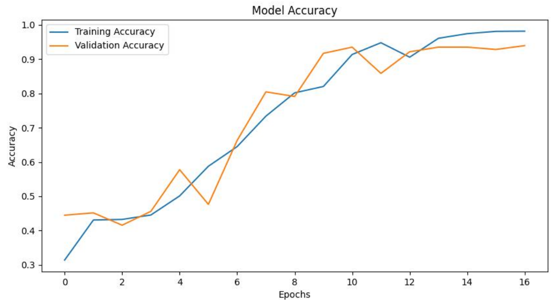
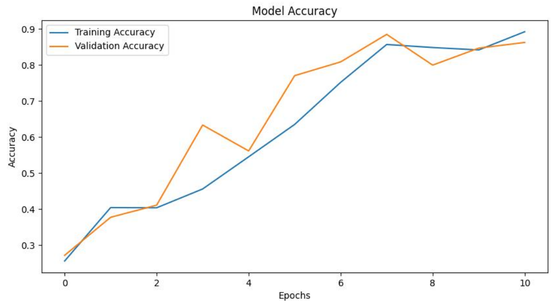
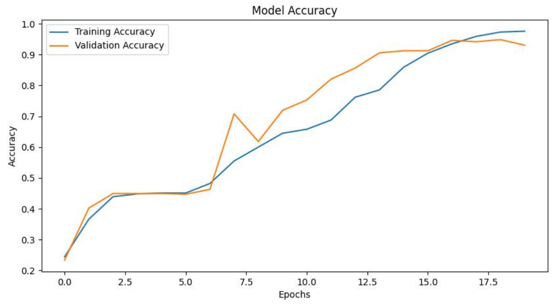
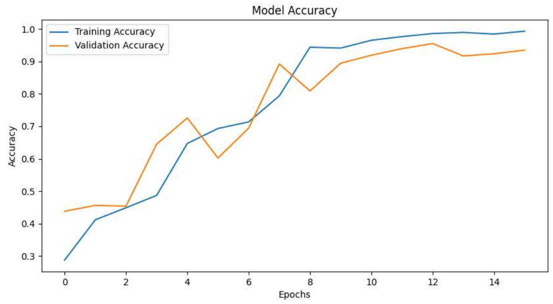
- **SVM**: C (regularization parameter) and kernel type (linear, RBF).
- **Logistic Regression**: C (Regularization Parameter) and Number of iterations.

For deep learning models, the following hyperparameters were tuned:

- **LSTM/GRU**: Number of epochs, dropout rates, and batch size were changed to find the highest accuracy.

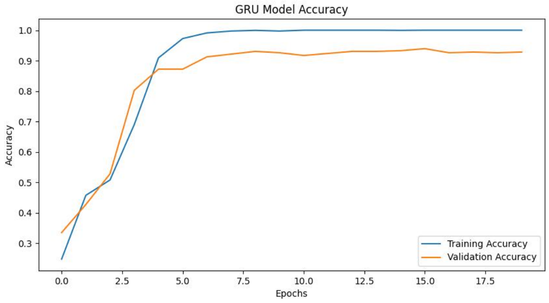
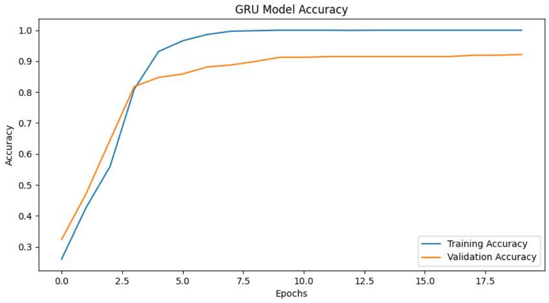
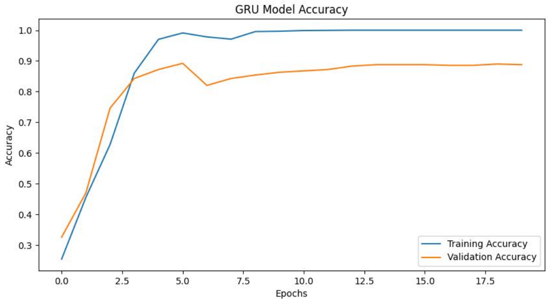
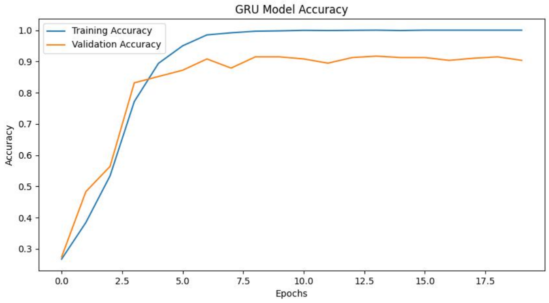
9.2.2. Hyperparameters Selection for LSTM

Table 1: Hyperparameters Selection for LSTM

Hyperparameters Selected	**Accuracy**	**Graph**
LSTM with dropout 0.5, batch size 32, and epochs 20	93.48%	
LSTM with dropout 0.3, batch size 64, and epochs 20	88.59%	
LSTM with dropout 0.5, batch size 128, and epochs 20	93.03%	
LSTM with dropout 0.5, batch size 32, and epochs 30	95.51%	

9.2.3. Hyperparameters Selection for GRU

Table 2: Hyperparameters Selection for GRU

Hyperparameters Selected	Accuracy	Graph
GRU with batch size 32, epochs 20 and embedding layers with 128 neurons	92.81%	
GRU with batch size 64, epochs 20 and embedding layers with 128 neurons	92.13%	
GRU with batch size 64, epochs 20 and embedding layers with 256 neurons	88.76%	
GRU with batch size 64, epochs 20 and embedding layers with 64 neurons	90.34%	

9.2.4. Regularization

For machine learning models, L2 regularization (via the `C` parameter in SVM and Logistic Regression) was used. In deep learning, dropout layers were incorporated to randomly deactivate neurons during training, preventing overfitting.

10. Evaluation Metrics and the Effect of Fine-Tuning

Each method and model implemented for text classification was evaluated using at least two metrics: **accuracy** and **F1-score**, ensuring a robust assessment of both overall performance and class-wise balance.

10.1. Machine Learning Models

1. **Logistic Regression**:
   - **Metrics**: Logistic Regression showed strong baseline performance due to its suitability for text classification.
   - **Effect of Fine-Tuning**: The regularization parameter was fine-tuned, leading to improved F1-scores across all categories by reducing noise sensitivity.
2. **Support Vector Machine (SVM)**:
   - **Metrics**: SVM achieved high performance, particularly in separating linearly separable data.
   - **Effect of Fine-Tuning**: Fine-tuning the regularization parameter (C) and kernel type resulted in a slight increase in accuracy and F1-scores.

10.2. Deep Learning Models

1. **LSTM**:
   - **Metrics**: LSTM excelled at capturing sequential dependencies, achieving high scores in most categories.
   - **Effect of Fine-Tuning**: Adjusting the number of hidden units and dropout rates improved generalization.
2. **GRU**:
   - **Metrics**: GRU performed comparably to LSTM, with slightly faster training times.
   - **Effect of Fine-Tuning**: Batch Size and dropout regularization were optimized, leading to improved accuracy and F1-scores.

10.3. Visualizations

1. **Confusion Matrices**:

Confusion Matrix for Logistic Regression

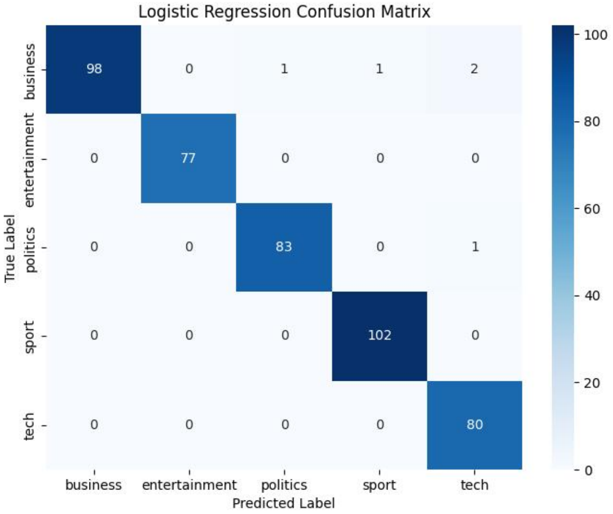


Figure 9: Confusion Matrix for Logistic Regression

Confusion Matrix for Support Vector Machine

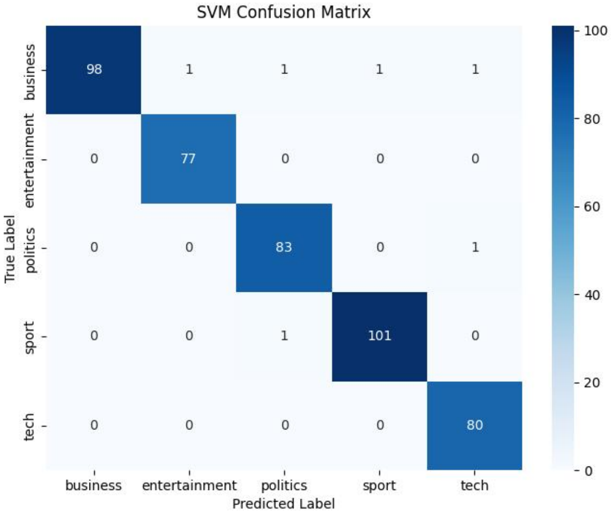


Figure 10: Confusion Matrix for Support Vector Machine

2. **Precision Recall Curves**:
   These curves provided insights into the trade-off between precision and recall, showcasing SVM's ability to maintain high precision at lower recall levels and Logistic Regression's balanced performance across varying thresholds.

Precision Recall Curve for Logistic Regression

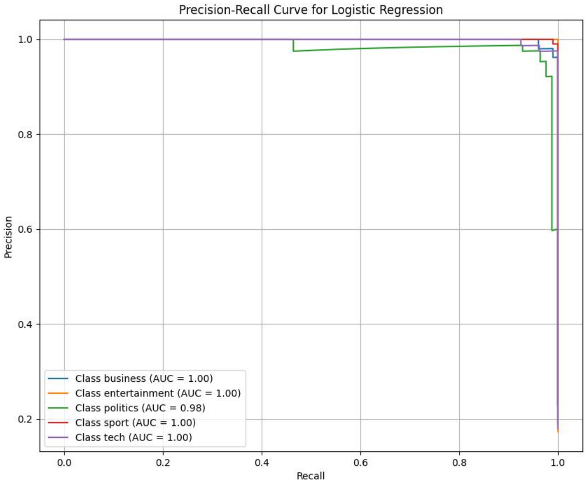


Figure 11: Precision Recall Curve for Logistic Regression

Precision Recall Curve for Support Vector Machine

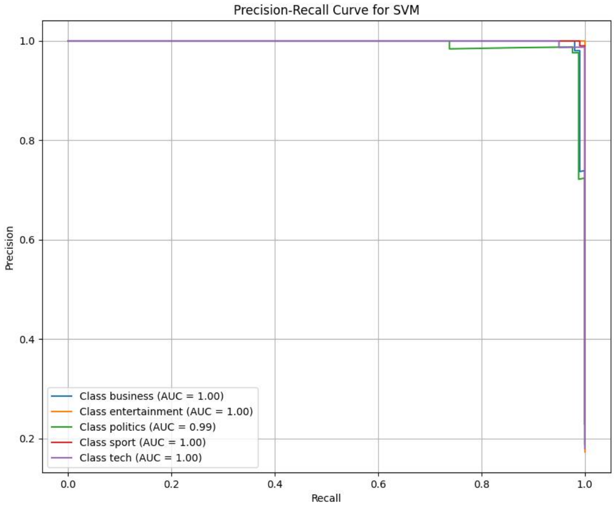


Figure 12: Precision Recall Curve for Support Vector Machine

11. Conclusion

This report provides an in-depth exploration of text classification, demonstrating a robust methodology and critical analysis to address the objectives outlined. Using Machine Learning and Deep Learning Algorithms, we successfully classified the news text to their appropriate categories.

The report employs the Harvard citation style, a widely-used author-date system that ensures clarity and ease of source attribution. Compared to MLA, which emphasizes in-text citations with page numbers for humanities research, or APA, which highlights publication years for social sciences, Harvard strikes a balance suitable for general academic contexts. Chicago style, with its detailed footnotes or endnotes, is ideal for historical research but may be cumbersome for technical reports. In contrast, Vancouver style, with its numerical references, is highly efficient for scientific fields but lacks the immediate context provided by author-date systems.

12. References

Amrita Bhattacharjee, R. M. J. G. H. L., 2024. Towards LLM-guided Causal Explainability for Black-box Text Classifiers. p. 6.

Anon., 2022. *Business Wire.* [Online]
Available at: http://www.businesswire.com/news/home/20220224005924/en/Natural-Language-Processing-NLP-Industry-Assessment-and-Forecast-2022-2027-by-Type-Technology-Deployment-Mode-Organization-Siz

Anon., 2022. *Global News Wire.* [Online]
Available at: https://www.globenewswire.com/en/news-release/2022/02/28/2392927/28124/en/Global-Natural-Language-Processing-NLP-Market-2022-2027-Widespread-Adoption-of-NLP-in-Healthcare-Call-Centers-and-Social-Media-Platforms.html

Christopher Schröder, G. H., 2024. Self-Training for Sample-Efficient Active Learning for Text Classification with Pre-Trained Language Models. p. 18.

Shervin Minaee, N. K. E. C., 2021. Deep Learning Based Text Classification: A Comprehensive Review. p. 43.

Zhen Li, X. W. W. Y. J. W. Z. Z. Z. L. M. S. H. Z. S. L., 2022. A Unified Understanding of Deep NLP Models for Text Classification. p. 15.

Zhiqiang Wang, Y. P. Y. L. X. Z., 2024. Adaptable and Reliable Text Classification using Large Language Models. p. 8.