Categorisation of Urdu News Articles

Amaan Ahmed
25100127@lums.edu.pk
Lahore University of Management Sciences
Lahore, Punjab, Pakistan

Aamil Khan Mahar
25020240@lums.edu.pk
Lahore University of Management Sciences
Lahore, Punjab, Pakistan

Ahmad Zubair Khan
25020232@lums.edu.pk
Lahore University of Management Sciences
Lahore, Punjab, Pakistan

Hamza Shakeel
25020116@lums.edu.pk
Lahore University of Management Sciences
Lahore, Punjab, Pakistan

Khan Abdullah Bin Azim
25020194@lums.edu.pk
Lahore University of Management Sciences
Lahore, Punjab, Pakistan

Abstract

The rise in accessibility of online news and articles in less prevalent languages such as Urdu demands an equivalent increase in the development of such informational content. While the content creation and compilation are handled adequately enough, automated classification methods would undoubtedly simplify and facilitate the organisation of articles into suitable categories on websites. This project seeks to identify the potential of such systems by creating and evaluating various models designed to classify Urdu news articles into separate categories. We collected over 1,000 articles from popular Urdu news websites in Pakistan, such as ARY News, Jang, Dunya, and Express. We then process and clean these raw articles, extracting the necessary features to implement three classification models, namely Multinomial Naïve Bayes, Logistic Regression, and Neural Networks. Our results show that the Naïve Bayes approach outshines with an accuracy of 95.20%, and is able to achieve the best results, while our logistic regression model (94.09%) is able to outperform our neural network model (93.16%). This paper outlines the methods applied, our results, and future development opportunities for optimising digital Urdu content distribution.

CCS Concepts

• **Computing methodologies** → **Neural networks**; **Classification and regression trees**; **Bayesian network models**.

Keywords

Urdu, Classification, Regression, Probabilistic Models, Machine Learning, Neural Networks, SKLearn, Naive Bayes, Text Classification, Data Cleaning, Scraping

ACM Reference Format:
Amaan Ahmed, Aamil Khan Mahar, Ahmad Zubair Khan, Hamza Shakeel, and Khan Abdullah Bin Azim. 2024. Categorisation of Urdu News Articles.

LUMS, Fall 2024, Lahore, Punjab, Pakistan

1 Introduction

With the advancement of technology, it has become increasingly simple to access a wide range of content with few simple clicks on the Internet. Similarly, the digital age has led to an exponential increase in demand for particular and readily available news content.

While major progress has been achieved for global languages such as English, Urdu remains underserved. Urdu, which is spoken by more than 230 million people, continues to lack innovation in tools and content categorisation. This gap requires users to actively sift through unstructured content, which reduces accessibility and user satisfaction. At the same time, it discourages people who want to consume news, or can only consume content, in Urdu from doing so digitally. Urdu's challenges, while unique, are something that most non-global languages face in this sphere: unstructured data, contextual nuances, and limited datasets, all of which hinder the development of automated solutions. Moreover, beyond the obvious benefits of personalization and accessibility, a robust classification system provides a foundation for advanced applications, such as real-time news recommendations and sentiment analysis, which can further enrich the user experience.

This project aims to address these points by developing an automated classification system that classifies Urdu news articles into preset categories: entertainment, business, sports, science and technology, and international, and as a minor step towards transforming unstructured data into ordered information, use popular machine learning methods to pave the path for smarter, more personalised news delivery systems.

2 Objectives

This project focuses on:

- Scraping Data: Collecting a dataset of over 1,000 Urdu news articles from platforms like Jang, Dunya, ARY News, and Express.
- Pre-processing Data: Process data for model implementation by cleaning, removing stopwords and structuring text data.
- Model Implementation: Implemented and evaluated three machine learning models:
  - Multinomial Naive Bayes: A probabilistic model for categorizing texts.
  - Logistic Regression: A regression model for multi-class classification.
  - Neural Networks: A neuron based non-linear model which explores complicated patterns within the sample texts, for multi-class classification.
- Model Evaluation: Evaluating different based on metrics like accuracy, precision, recall, and F1-score and find the best-performing model. Models created were also evaluated in performance against SKLearn models.

3 Methodology

This section describes the process taken to create three Urdu news classification systems. Our method starts by compiling and scraping a collection of Urdu news articles from various web sources. The data was then thoroughly cleaned to ensure its quality and suitability for our intended models. We then implement said models, and perform training on this dataset, wherever necessary. Each stage of our process is explained in detail in the subsections to follow.

3.1 Data Scraping and Collection

Given the limited availability of Urdu datasets, particularly in the context of news articles, we undertook the task of creating a comprehensive, up-to-date dataset to align with our objective of developing a robust system that addresses modern needs. To construct this dataset, news articles were collected from prominent Urdu news websites, including ARY News [2], Dunya News [3], Express News [4], and Jang News [5]. Custom web scraping scripts were developed using Python libraries, such as BeautifulSoup and Selenium, to systematically extract article content, titles, and categories from the structured HTML pages of these websites.

As each site has unique layouts and categorisation, the dataset was then standardised according to our pre-set categories; Entertainment, International, Sports, Business, and Science and Technology, to streamline the collection of ground-truth labels, facilitating efficient model evaluation during later stages of our paper. Articles were retrieved from the preceding weeks at the time of collection, ensuring the dataset's relevance to contemporary events. Measures were implemented to eliminate duplicate entries and filter out irrelevant data. The web scraping scripts used for this process are publicly available in our GitHub repository, providing transparency and a framework for reproducibility, for future research efforts.

3.2 Data Preprocessing and Cleaning

After the raw Urdu news articles were compiled, the dataset was thoroughly cleaned to prepare it for later use in our models. The first step was to remove stopwords using a publicly available Urdu stopwords dataset [1]. Due to the fact that different source websites had variations in their layouts and category schemes, the collected articles' ground-truth labels were standardized to align with the preset categories defined for this study. Duplicate entries and records with missing values were systematically removed, and the remaining HTML tags or non-Urdu alphanumeric characters were removed to increase the data consistency and integrity. The cleaned articles were aggregated into one dataframe and used for model implementation and evaluation. Scripts for cleaning and dataset compilation used are included in our repository.

3.3 Model Implementation

This section outlines the implementation of three different models in the task of classifying Urdu news articles: Multinomial Naive Bayes, Logistic Regression, and a basic Neural Network. The choice of these models was based on their varied degrees of complexity and relation to tasks with text classification. Every model was designed from scratch to enable better understanding of their working processes; and, the results were benchmarked against SKLearn's existing implementations to evaluate their effectiveness.

3.3.1 ***Multinomial Naive Bayes.*** Multinomial Naive Bayes is a probabilistic model often used in text classification tasks due to its simplicity and effectiveness when features are discrete, such as word counts. We implemented the algorithm from scratch, calculating the probabilities of each feature given a class, and then used these probabilities to classify new articles. For comparison, we also

utilized SKLearn's built-in Multinomial Naive Bayes implementation, which streamlined the process with optimized performance. The comparison enabled us to assess the impact of our manual work versus the efficiency of SKLearn's implementation.

3.3.2 ***Logistic Regression and Multi-class Logistic Regression.*** Logistic Regression, both in its binary and multi-class forms, was built from scratch to address the classification of Urdu news articles. We built the model by specifying the sigmoid and cost functions for optimisation. For multi-class classification, we used the one-versus-rest technique. Additionally, we used SKLearn's logistic regression implementation to assess our results, verifying that the performance of our tailored implementation met the standards.

3.3.3 ***Neural Network.*** We used a basic feed-forward neural network for text classification as a more advanced approach. An input layer, hidden layers, and an output layer, with cross-entropy loss, to manage multi-class predictions made up the network. Gradient descent and backpropagation were used to train the model. The Multi-layer Perceptron (MLP) implementation from SKLearn, offered a strong framework for comparison. We were able to examine the trade-offs between a scratch implementation and a high-level framework for managing non-linear relationships in text data by comparing our custom neural network with the SKLearn implementation.

4 Results

As previously mentioned, three distinct machine learning models were applied to the dataset collected through scraping. These models include **Multinomial Naive Bayes**, **Feed Forward Neural Network (FFNN)**, and **Logistic Regression**. Each model was trained and evaluated based on their ability to accurately classify data into predefined categories. The results of these models are summarized as follows:

4.1 Accuracy

Accuracy is one of the most widely used metrics for classification tasks, and it represents the overall proportion of correctly classified instances in the dataset. In this study, three machine learning models were evaluated, and their accuracy scores were as follows:

- **Logistic Regression**: This model achieved an accuracy of **94.09%**, demonstrating its effectiveness in correctly classifying instances. Logistic Regression is particularly well-suited for linear classification tasks and performed well on this dataset, where the relationships between the features and classes may have been relatively linear. The accuracy achieved using the *scikit-learn* (sklearn) implementation was **93.47%**.
- **Feed Forward Neural Network (FFNN)**: Despite being a more complex model capable of capturing intricate patterns, the FFNN achieved an accuracy of **93.16%**. This result suggests that, while FFNNs are more flexible, they may require more tuning, hyperparameter adjustments, and better data preprocessing to outperform simpler models like Logistic Regression in this particular task. The accuracy of the FFNN using the *scikit-learn* (sklearn) implementation was **94.4%**.
- **Multinomial Naive Bayes**: The Multinomial Naive Bayes classifier outperformed the other models, achieving an accuracy of **95.20%**. Despite its simplicity and the assumption of feature independence, Naive Bayes provided the highest accuracy, highlighting the effectiveness of probabilistic models when dealing with categorical or structured features. This strong performance could also be attributed to the model's efficient handling of features and its ability to classify based on probabilistic reasoning. The accuracy for Naive Bayes using *scikit-learn* (sklearn) was **95.24%**, matching the custom implementation.

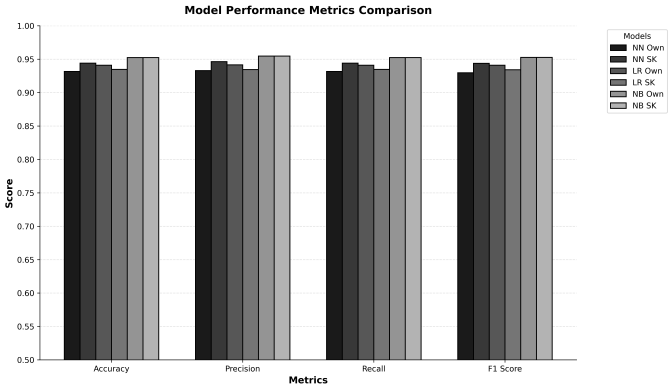

Figure 1: Accuracy, Precision, Recall and F1 Score of the models.

4.2 Precision

Precision is a metric that measures the proportion of correctly predicted positive observations out of all predicted positives. It is especially valuable in situations where the cost of false positives is high, such as in fraud detection or medical diagnoses. Below are the precision values for the three machine learning models, considering both their custom implementations and the *scikit-learn* versions:

- **Logistic Regression**: The precision achieved by the Logistic Regression model using the *own implementation* was **94.14%**. This indicates that, among all the positive predictions made by the model, 94.14% were correct. In contrast, when the model was implemented using *scikit-learn*, the precision was slightly lower at **93.47%**. The small difference in precision suggests that both implementations performed similarly, with the sklearn version perhaps being more robust due to its optimizations.
- **Feed Forward Neural Network (FFNN)**: The precision of the Feed Forward Neural Network using the *own implementation* was **93.29%**. This indicates that 93.29% of the predictions that the model classified as positive were indeed correct. Using *scikit-learn*, the FFNN achieved a higher precision of **94.63%**. This improvement suggests that the sklearn version of FFNN benefits from additional fine-tuning and optimizations, leading to better performance in this regard.

- **Multinomial Naive Bayes**: The Multinomial Naive Bayes classifier achieved a precision of **95.46%** with the *own implementation*, meaning that around **95%** of its positive predictions were accurate. Interestingly, when the model was implemented using *scikit-learn*, the precision was exactly **95.46%**.

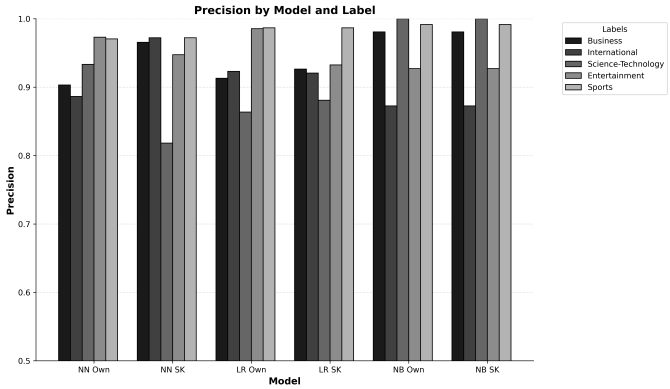

Figure 2: Precision scores for each label for each model.

The accompanying **Figure 2** visualizes the **precision** values for each label across all three types of models. It shows how each model performs in terms of **precision** for individual categories, helping to further understand where each model excels or may require improvements.

4.3 Recall

Recall, also known as Sensitivity or True Positive Rate, measures the proportion of actual positive instances that are correctly identified by the model. This metric is particularly critical when it is important to minimize false negatives, as it highlights the model's ability to capture as many relevant instances as possible. Below are the recall scores for each model:

- **Logistic Regression**: The recall achieved by the Logistic Regression model was **94.09%** for the own implementation and **93.47%** when using *scikit-learn*. Both implementations demonstrated a strong ability to identify positive instances. The slight decrease in recall when using the *scikit-learn* version can be attributed to differences in how the models are trained and optimized. Logistic Regression is generally effective for tasks where the relationships between the features and the target are linear, and both implementations were able to capture a large portion of positive instances with high recall scores.
- **Feed Forward Neural Network (FFNN)**: The recall achieved by the Feed Forward Neural Network was **93.16%** for the own implementation and **94.41%** when using *scikit-learn*. Interestingly, the *scikit-learn* implementation provided a higher recall, indicating that the neural network was better able to detect positive instances in this setup. This may be due to more efficient training techniques, better hyperparameter optimization, or different initialization schemes used in *scikit-learn*. The FFNN, being a more complex model with multiple layers, has the ability to capture intricate patterns in data, which contributed to its strong recall performance, especially in the *scikit-learn* version.
- **Multinomial Naive Bayes**: The recall achieved by the Naive Bayes model was **95.23%** for both the own implementation and the *scikit-learn* implementation. This consistency across both versions indicates that Multinomial Naive Bayes was highly effective in identifying positive instances. The Multinomial Naive Bayes model is particularly strong in scenarios where the assumption of feature independence holds true, and it was able to achieve an impressive recall score in this task. Even with the relatively simple nature of Naive Bayes, its performance remained robust across different implementations, making it a strong candidate for high-recall tasks in this dataset.

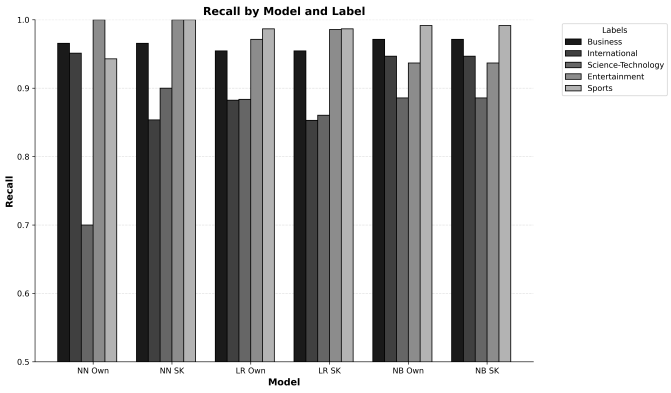

Figure 3: Recall scores for each label for each model.

The accompanying **Figure 3** visualizes the **recall** values for each label across all three types of models. It shows how each model performs in terms of **recall** for individual categories, helping to further understand where each model excels or may require improvements.

4.4 F1 Score

The F1 Score is the harmonic mean of precision and recall, providing a balanced measure of a model's ability to correctly identify positive instances while avoiding false positives and false negatives. It is particularly useful when the dataset is imbalanced, as it gives a more nuanced view of the model's performance. Below are the F1 scores for each model:

- **Logistic Regression**: The F1 score achieved by the Logistic Regression model was **94.10%** for the own implementation and **93.41%** for the *scikit-learn* version. The F1 score remained relatively consistent across both implementations, suggesting that Logistic Regression effectively balanced precision and recall. Despite the slight drop in F1 score with the *scikit-learn* version, both implementations demonstrated solid performance in capturing relevant positive instances while maintaining a low rate of false positives and false negatives.

- **Feed Forward Neural Network (FFNN)**: The F1 score for the Feed Forward Neural Network was **92.96%** for the own implementation and **94.37%** for the *scikit-learn* implementation. The *scikit-learn* version showed a notable improvement in the F1 score, reflecting its enhanced ability to balance both precision and recall. This suggests that, while the FFNN was capable of capturing complex patterns in the data, the *scikit-learn* implementation's optimizations helped it achieve a better balance between precision and recall, leading to improved overall performance.
- **Multinomial Naive Bayes**: The Multinomial Naive Bayes model achieved an F1 score of **95.27%** for both the own implementation and the *scikit-learn* version. This consistency across both versions highlights the strength of Naive Bayes in balancing precision and recall, even with its simplicity. Its ability to maintain a high F1 score indicates that it was able to correctly classify instances without excessively favoring precision or recall, making it a reliable model for this dataset.

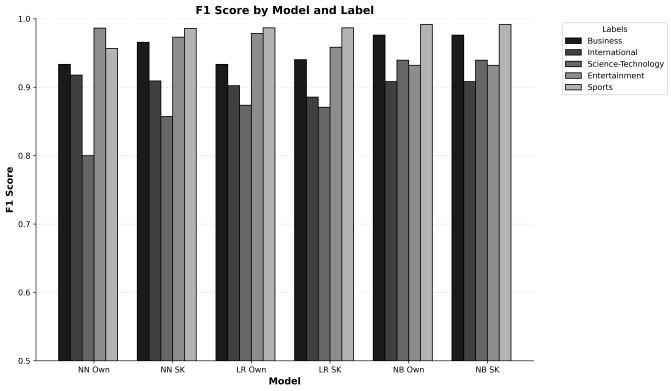

Figure 4: F1 scores for each label for each model.

The accompanying **Figure 4** visualizes the **f1 score** values for each label across all three types of models. It shows how each model performs in terms of **f1 score** for individual categories, helping to further understand where each model excels or may require improvements.

4.5 Classification Reports

In this section, we present the detailed classification reports for each of the six models evaluated in this study. These models include both custom implementations and the corresponding *scikit-learn* versions: Logistic Regression, Feed Forward Neural Network (FFNN), and Multinomial Naive Bayes.

Following the presentation of these reports, we will discuss the overall findings, comparing the performance of the models and drawing conclusions based on their strengths and weaknesses. This analysis will help us understand which model is best suited for this particular classification task and provide guidance for future research and model selection.

Metric	Precision	Recall	F1-Score
Business	0.91	0.95	0.93
International	0.99	0.97	0.98
Science-Technology	0.92	0.88	0.90
Entertainment	0.86	0.88	0.87
Sports	0.99	0.99	0.99
Accuracy		0.94	
Macro Avg	0.93	0.94	0.93
Weighted Avg	0.94	0.94	0.94

Table 1: Logistic Regression: Own Classification Report

The custom implementation of Logistic Regression achieved high overall accuracy of 94%, with excellent performance in categories like "International" and "Sports." The model struggled slightly with "Entertainment" and "Science-Technology," where the F1-scores were lower, indicating potential room for feature engineering or hyperparameter tuning. Despite these challenges, the implementation performed robustly across most metrics.

Metric	Precision	Recall	F1-Score
Business	0.93	0.95	0.94
International	0.93	0.99	0.96
Science-Technology	0.92	0.85	0.89
Entertainment	0.88	0.86	0.87
Sports	0.99	0.99	0.99
Accuracy		0.93	
Macro Avg	0.93	0.93	0.93
Weighted Avg	0.93	0.93	0.93

Table 2: Logistic Regression: SKLearn Classification Report

The *scikit-learn* implementation of Logistic Regression achieved an accuracy of 93%, slightly lower than the custom implementation. The model showed strong performance in "International" and "Sports" but weaker results for "Science-Technology," with a recall of 0.85. The consistency of the results highlights the reliability of Logistic Regression across implementations, though additional optimization may improve performance for challenging categories.

Metric	Precision	Recall	F1-Score
Business	0.98	0.97	0.98
International	0.93	0.94	0.93
Science-Technology	0.87	0.95	0.91
Entertainment	1.00	0.89	0.94
Sports	0.99	0.99	0.99
Accuracy		0.95	
Macro Avg	0.95	0.95	0.95
Weighted Avg	0.95	0.95	0.95

Table 3: Multinomial Naive Bayes: Own Classification Report

The custom implementation of Multinomial Naive Bayes achieved the highest accuracy of 95%, with consistently strong precision, recall, and F1-scores across categories. The model was particularly

effective in "Entertainment" and "Sports," where near-perfect scores were observed. Its simplicity and probabilistic approach made it an efficient and reliable model for this dataset.

Metric	Precision	Recall	F1-Score
Business	0.98	0.97	0.98
International	0.93	0.94	0.93
Science-Technology	0.87	0.95	0.91
Entertainment	1.00	0.89	0.94
Sports	0.99	0.99	0.99
Accuracy		0.95	
Macro Avg	0.95	0.95	0.95
Weighted Avg	0.95	0.95	0.95

Table 4: Multinomial Naive Bayes: SKLearn Classification Report

The *scikit-learn* implementation of Multinomial Naive Bayes mirrored the custom implementation, with an accuracy of 95% and excellent metrics across all categories. The minor differences in individual category performance emphasize the robustness of Multinomial Naive Bayes as a simple yet effective model for categorical datasets.

Metric	Precision	Recall	F1-Score
Business	0.90	0.97	0.93
International	0.97	1.00	0.99
Science-Technology	0.89	0.95	0.92
Entertainment	0.93	0.70	0.80
Sports	0.97	0.94	0.96
Accuracy		0.93	
Macro Avg	0.93	0.91	0.92
Weighted Avg	0.93	0.93	0.93

Table 5: Neural Network: Own Classification Report

The custom implementation of the Feed Forward Neural Network achieved an accuracy of 93%, demonstrating strong performance in most categories. However, "Entertainment" exhibited lower recall and F1-scores, which suggests the model struggled with imbalanced or nuanced data. This highlights the importance of hyperparameter tuning and additional training for improving neural network performance.

Metric	Precision	Recall	F1-Score
Business	0.97	0.97	0.97
International	0.95	1.00	0.97
Science-Technology	0.97	0.85	0.91
Entertainment	0.82	0.90	0.86
Sports	0.97	1.00	0.99
Accuracy		0.94	
Macro Avg	0.94	0.94	0.94
Weighted Avg	0.95	0.94	0.94

Table 6: Neural Network: SKLearn Classification Report

The *scikit-learn* implementation of the Feed Forward Neural Network achieved an improved accuracy of 94%, showing balanced precision and recall across most categories. "Entertainment" improved compared to the custom implementation, though it remained relatively weak compared to other categories. This suggests the *scikit-learn* implementation benefited from additional optimizations.

5 Conclusion

Among all models, Multinomial Naive Bayes achieved the highest accuracy of 95% across both implementations, demonstrating strong performance and consistency. Logistic Regression provided competitive results with slightly lower accuracy but excellent metrics in specific categories. The Feed Forward Neural Network showed potential for capturing complex patterns but required more tuning to outperform simpler models. These findings highlight the trade-offs between model complexity, optimization, and dataset characteristics in machine learning classification tasks.

6 Limitations

While the proposed Urdu news classification system demonstrates promising results, there are several limitations that must be acknowledged. One of the primary constraints is the relatively small dataset used for model training. The dataset consists of articles from a limited number of news sources, which restricts the diversity of topics and writing styles, potentially affecting the generalization ability of the models. Expanding the dataset by incorporating a broader range of articles from various news outlets would help address this limitation.

Additionally, the classification system is currently limited to only five predefined categories: Entertainment, International, Sports, Business, and Science-Technology. This narrow focus might not capture the full spectrum of topics found in Urdu news articles. Incorporating more granular or additional categories could improve the model's versatility in handling a wider variety of news content.

Moreover, the complexity of the Urdu language presents its own challenges. The subtle nuances, variations in vocabulary, and diverse syntactic structures in Urdu can lead to misclassification or ambiguity in some cases. The use of a public Urdu stopwords dataset, although helpful, could also be enhanced. A more comprehensive and domain-specific stopwords list could improve the text preprocessing stage, reducing noise and ensuring better model performance. These limitations highlight the need for further improvements in data quality, model complexity, and linguistic adaptation for a more robust and accurate classification system.

7 Future Work

The Urdu news classification system can be further improved and enhanced in a number of ways in the future. To improve the classification system's performance, one of the main approaches is to investigate and use more models. More sophisticated models like Support Vector Machines (SVM), Random Forests, and ensemble approaches could offer substantial improvements in classification accuracy beyond the models examined in this paper. Furthermore, using deep learning models such as Transformer-based architectures like BERT or GPT, could increase the model's comprehension of contextual subtleties and accuracy, particularly when working with the dense language structures inherent in Urdu.

Another important area for future research is dataset expansion. A larger and more diversified dataset might be required to improve the classification system's adaptability and generalisation. Currently, the dataset used is confined to a few well-known Urdu news websites. Expanding this dataset to include articles from a broader range of Urdu media outlets would not only give more diverse training data, but will also improve the model's ability to categorise articles from various sources. Furthermore, integrating articles in other Pakistani regional languages, such as Pashto, Punjabi, and Sindhi, could broaden the system's appeal to a wider linguistic audience, resulting in a more inclusive and comprehensive news classification tool for Pakistan.

Finally, to improve the system's accessibility and usability, an open-source application that simply integrates this classification model into websites could be created. By developing a simple, plug-and-play solution, news websites and content providers may easily utilise the system to automatically identify and categorise articles, thus improving the content management process by ensuring that articles are properly tagged for readers. This tool can also provide an API, allowing developers and organisations to integrate the classification system into their own platforms, promoting widespread adoption and use.

References

[1] Publicly Available Urdu Stopwords Dataset. 2024. Urdu Stopwords Dataset. https://github.com/stopwords-iso/stopwords-ur.
[2] ARY News. 2024. ARY News. https://urdu.arynews.tv.
[3] Dunya News. 2024. Dunya News. https://urdu.dunyanews.tv.
[4] Express News. 2024. Express News. https://www.express.pk.
[5] Jang News. 2024. Jang News. https://jang.com.pk/.

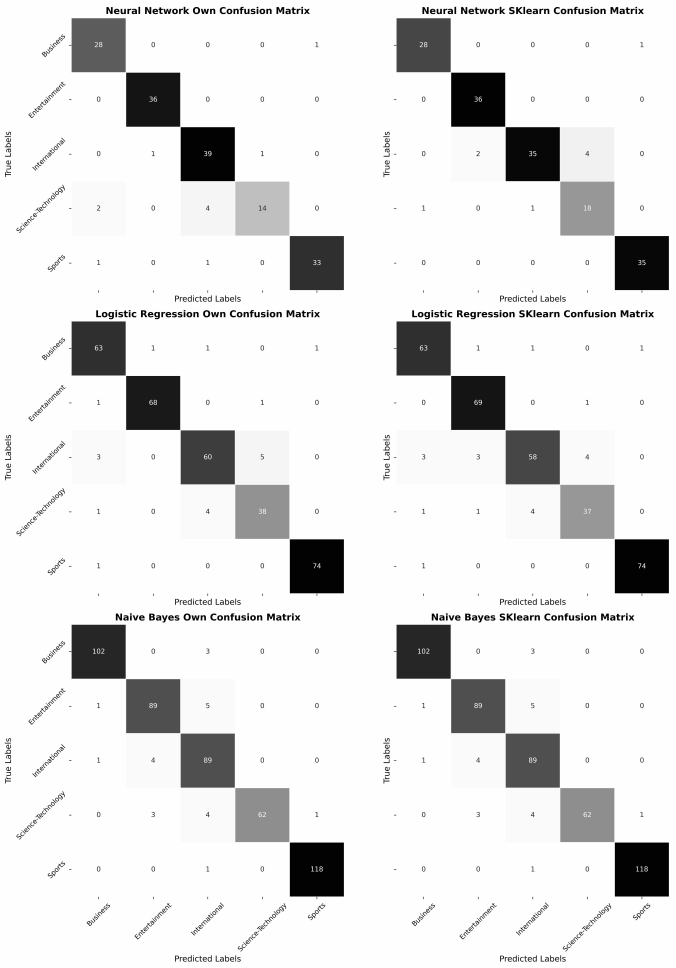

Figure 5: Confusion matrices.

8 Repository Link

GitHub repository: link. All results, graphs and scripts can be found in this repository.