Multiclass Text Classification of Urdu News Articles Using Machine Learning

Ahmad Junaid, Ayesha Khaldoon, Talha Rehan, Sheraz Waseem
Lahore University of Management Sciences
Lahore, Pakistan

ABSTRACT

With the rise of digital media, the global proliferation of digital news necessitates computational tools to efficiently process and analyze vast volumes of data. While substantial progress has been made in classifying news in languages like English, Urdu—a widely spoken language in the subcontinent remains underexplored due to limited resources. This paper addresses the challenge of multiclass text classification applied to news articles in the Urdu language. The project involved scraping Urdu news articles from multiple prominent news websites, followed by cleaning and tokenizing the content. Three different models were implemented and evaluated for their ability to classify the articles into five distinct categories based on the gold labels: Entertainment, Business, Sports, Science-Technology, and International. A fully connected neural network (FCNN), logistic regression, and Multinomial Naive Bayes model were implemented from scratch for comparison. The results indicate that deep learning models demonstrate strong performance for text classification tasks in non-English languages. Key challenges encountered during data preprocessing, model training, and optimization are discussed, alongside insights into potential future research directions in this domain.

ACM Reference Format:
Ahmad Junaid, Ayesha Khaldoon, Talha Rehan, Sheraz Waseem . . Multiclass Text Classification of Urdu News Articles Using Machine Learning. In *Proceedings of* . ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Nowadays, information on the Internet is available in a range of simple to understand spoken languages. Text classification is a fundamental task in NLP that has seen widespread application across industries, particularly for organizing and retrieving textual data. In the Indo-Pak region, Urdu serves as Pakistan's official language, with approximately 220 million speakers in Pakistan and over 500 million worldwide. Despite its wide usage, Urdu remains underexplored in computational linguistics making automated classification challenging compared to languages like English which benefit from extensive annotated datasets and advanced pre-trained models. Urdu's rich morphology and diverse vocabulary, derived primarily from Arabic and Persian, further complicate the task.This paper focuses on automating the classification of Urdu news articles into predefined categories. Building on prior works that applied machine learning to English text classification, this paper examines the performance of Logistic Regression, Multinomial Naive Bayes, and Fully Connected Neural Networks (FCNN) on Urdu datasets. Section 2 details the methodology, including data collection, preprocessing, tokenizing and model implementations. Section 3 presents the Models. Section 4 discusses the results, and Section 5 concludes the study with future considerations.

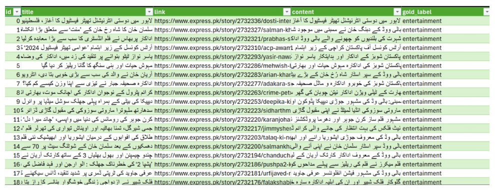
, ,
.

2 METHODOLOGY

2.1 Data Collection

To build a comprehensive dataset of Urdu news articles, a custom Python-based scraper was developed. This scraper extracts articles from multiple prominent Urdu news websites, including Geo Urdu, Jang, ARY News, Dunya News, and Express News. The scraping process targeted five predefined categories: Entertainment, Business, Sports, Science-Technology, and International which will serve as our gold labels. The scraping process for each article involved:

- **Title Extraction:** Titles were retrieved from specific HTML tags, with additional cleaning steps to remove unnecessary spaces and special characters.
- **Content Retrieval:** Article content was gathered by parsing, which were then concatenated into cohesive paragraphs to maintain context and readability.
- **Link Collection:** URLs for each article were recorded for traceability and potential validation.
- **Gold Labels:** Categories were automatically assigned as "gold labels" based on the section from which the article was scraped.



id	title	link	content	gold_label
0	لاہور میں دوستی انٹرنیشنل تھیٹر فیسٹیول کا آغاز، فلسطینیو	https://www.express.pk/story/2732336/dosti-inter	لاہور میں دوستی انٹرنیشنل تھیٹر فیسٹیول کا آغاز	entertainment
1	سلمان خان کا شاہ رخ خان کے 'منت' سے متعلق بڑا انکشا	https://www.express.pk/story/2732327/salman-kh	بالی ووڈ کے دبنگ خان نے ممبئی میں موجود شا	entertainment
2	اداکار پربھاس نے فلم انڈسٹری کا سب سے بڑا معاہدہ کرلیا	https://www.express.pk/story/2732321/prabhas-s	شہرت کی بلندیوں کو چھونے والے بالی ووڈ اداک	entertainment
3	آرٹس کونسل کے زیر اہتمام 'عوامی تھیٹر فیسٹیول 2024' ڈ	https://www.express.pk/story/2732310/acp-awam	آرٹس کونسل آف پاکستان کراچی کے زیر اہتمام	entertainment
4	یاسر نواز ٹیٹو بنوانے پر تنقید کی زد میں، اداکار کی وضاح	https://www.express.pk/story/2732293/yasir-naw	پاکستان شوبز کے اداکار اور ہدایتکار یاسر نواز	entertainment
5	مہوش حیات اور بنی سنگھ کا گانا ریلیز کر دیا گیا	https://www.express.pk/story/2732286/mehwish-l	پاکستان شوبز کی اداکارہ مہوش حیات اور بھارتی	entertainment
6	آریان خان نے والد کی سب سے بڑی خوبی بتا دی، انٹرویو	https://www.express.pk/story/2732283/arian-khar	بالی ووڈ کے سپر اسٹار شاہ رُخ خان کے بڑے بیٹ	entertainment
7	اداکارہ صحیفہ جبار نے تیزی سے اپنا وزن کیسے کم کیا؟	https://www.express.pk/story/2732277/adakara-s	پاکستان شوبز کی خوبرو اداکارہ و ماڈل صحیفہ ج	entertainment
8	کرائم پٹرول کے نوجوان اداکار کی اچانک موت، بھارتی اد	https://www.express.pk/story/2732263/crime-pet	بھارت کے ٹیلی ویژن اداکار نیتن چوہان کی گھر	entertainment
9	دیپیکا کی بیٹی کے ہمراہ پہلی جھلک سوشل میڈیا پر وائرل	https://www.express.pk/story/2732253/deepika-ki	ممبئی: بالی وڈ کی مشہور جوڑی دیپیکا پڈوکون او	entertainment
10	سدھارتھ ملہوترا ماروتی سوزوکی کی مقبول گاڑی ڈرائیر ک	https://www.express.pk/story/2732223/sidharthm	ماروتی سوزوکی انڈیا لمیٹڈ نے اپنی مقبول گاڑی	entertainment
11	کرن جوہر کی رومانس کی دنیا میں واپسی، 'چاند میرا دل'	https://www.express.pk/story/2732220/karanjoha	مشہور فلم ساز کرن جوہر اور دھرما پروڈکشنز	entertainment
12	جمی شیرگل، تمنا بھاٹیہ اور اویناش تیواری کی تھرلر فلم '	https://www.express.pk/story/2732217/jimmysher	نیٹ فلکس کی بہت انتظار کی جانے والی کرائم ڈ	entertainment
13	طلاق کی افواہوں کے درمیان ایشوریا اور ابھیشیک نئی فلم	https://www.express.pk/story/2732203/talaq-ki-ne	بالی وڈ کی معروف جوڑی ایشوریا رائے اور ابھ	entertainment
14	دھمکیوں کے بعد سلمان خان کے شوٹنگ سیٹ پر 70 سے	https://www.express.pk/story/2732200/salmankh	بالی ووڈ سپر اسٹار سلمان خان نے اپنی آنے والی	entertainment
15	چندو چیمپئن اور بھول بھلیاں 3 کے ساتھ کارتک آریان نے	https://www.express.pk/story/2732194/chanduch	بالی ووڈ کے معروف اداکار کارتک آریان کے لی	entertainment
16	'پشپا 2' کی خطرناک جھلک : الو ارجن اور فہد فاضل کی	https://www.express.pk/story/2732186/pushpa2-k	فلم میکرز نے فلم کی ریلیز سے پہلے مداحوں کو	entertainment
17	عرفی جاوید کی ٹرپتی ڈمری پر شدید تنقید، ڈانس سیکھنے ک	https://www.express.pk/story/2732181/urfijaved-r	بالی ووڈ کی مشہور فیشن انفلوئنسر عرفی جاوید	entertainment
18	فلک شبیر نے ازدواجی زندگی خوشگوار بنانے کا راز بتا د	https://www.express.pk/story/2732176/falakshabi	گلوکار فلک شبیر اور ان کی اہلیہ اداکارہ سارہ	entertainment

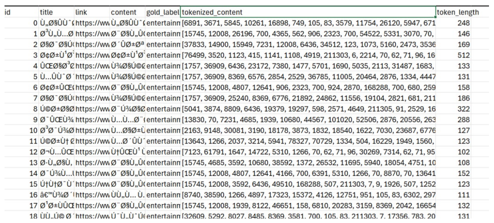
Figure 1: Dataset

2.2 Data Preprocessing

After scraping, the next step was Cleaning. To ensure data quality, duplicates were removed using a unique combination of title and content. Articles missing essential fields such as title, content, or labels were excluded, and text normalization was applied to remove HTML artifacts, extra spaces, and URLs using regular expressions, while standardizing Urdu text. To further refine our input, a predefined list of Urdu stopwords (STOP_WORDS) was used to filter out common but non-informative words, reducing noise in the data.[1]

Since our models cannot directly work on raw data, tokenization —a process of converting text into numerical representations was essential for making the cleaned text comprehensible to a computer program. To achieve this, the GPT-2-based Urdu tokenizer (hadidev/gpt2-urdu-tokenizer-withgpt2) was used, which is specifically tailored for processing Urdu text [2]. This choice of tokenizer was made based on multiple advantages: unlike general-purpose tokenizers, this model was specifically trained on Urdu text, ensuring it captures linguistic nuances, unique grammar structures, and idiomatic expressions of the language. Moreover, it was already fine-tuned on Urdu datasets, which ensured robust performance for our main purpose—classification. Additionally, the GPT-2 Urdu tokenizer uses a subword tokenization approach, which breaks down words into meaningful segments. This reduces vocabulary size and handles out-of-vocabulary (OOV) words effectively, making it particularly suitable for morphologically rich languages like Urdu.

With the cleaned and tokenized dataset prepared, the next step involved leveraging multiple machine learning models to classify Urdu news articles into their respective categories and evaluating results.



id	title	link	content	gold_label	tokenized_content	token_length
0	Ù„Ø§ÛÙˆ	https://ww	Ù„Ø§ÛÙˆ	entertainm	[6891, 3671, 5845, 10261, 16898, 749, 105, 83, 3579, 11754, 26120, 5947, 671	248
1	Ø³Ù„Ù…Ø	https://ww	Ø¨Ø§Ù„Û	entertainm	[15745, 12008, 26196, 700, 4365, 562, 906, 2323, 700, 54522, 5331, 3070, 70,	146
2	Ø§Ø¯Ø§Ú	https://ww	Ø´ÛØ±Øª	entertainm	[37833, 14900, 15949, 7231, 12008, 6436, 34512, 123, 1073, 5160, 2473, 3536	169
3	Ø¢Ø±Ù¹Ø	https://ww	Ø¢Ø±Ù¹Ø	entertainm	[76499, 3520, 1123, 415, 1141, 1108, 4919, 211303, 6, 2214, 70, 62, 71, 96, 16	512
4	ÛŒØ§Ø³Ø	https://ww	Ù¾Ø§Ú©Ø	entertainm	[1757, 36909, 6436, 23172, 7380, 1477, 5701, 1690, 5035, 2113, 31487, 1683,	133
5	Ù…ÛÙˆØ	https://ww	Ù¾Ø§Ú©Ø	entertainm	[1757, 36909, 8369, 6576, 2854, 2529, 36785, 11005, 20464, 2876, 1334, 4447	131
6	Ø¢Ø±ÛŒØ	https://ww	Ø¨Ø§Ù„Û	entertainm	[15745, 12008, 4807, 12641, 906, 2323, 700, 924, 2870, 168288, 700, 680, 259	158
7	Ø§Ø¯Ø§Ú	https://ww	Ù¾Ø§Ú©Ø	entertainm	[1757, 36909, 25240, 8369, 6776, 21892, 24862, 11556, 19104, 2821, 681, 211	186
8	Ú©Ø±Ø§Ø	https://ww	Ø¨Ù¾Ø§Ø	entertainm	[5041, 3874, 8809, 6436, 19379, 19297, 598, 2571, 4649, 211305, 91, 2529, 16	322
9	Ø¯ÛŒÙ¾	https://ww	Ù…Ù…Ø¨	entertainm	[13830, 70, 7231, 4685, 1939, 10680, 44567, 101020, 52506, 2876, 20556, 263	288
10	Ø³Ø¯Ù¾Ø	https://ww	Ù…Ø§Ø±Ù	entertainm	[2163, 9148, 30081, 3190, 18178, 3873, 1832, 18540, 1622, 7030, 23687, 6776	127
11	Ú©Ø±Ù† Ø	https://ww	Ù…Ø´ÛÙˆ	entertainm	[13643, 1266, 2037, 3214, 5941, 78327, 70729, 1334, 504, 16229, 1949, 1560,	123
12	Ø¬Ù…ÛŒ	https://ww	Ù†ÛŒÙ¹ Ù	entertainm	[7123, 61791, 1647, 14722, 5310, 1266, 70, 62, 71, 96, 30269, 7314, 62, 71, 95	102
13	Ø·Ù„Ø§Ù‚	https://ww	Ø¨Ø§Ù„Û	entertainm	[15745, 4685, 3592, 10680, 38592, 1372, 26532, 11695, 5940, 18054, 4751, 10	108
14	Ø¯Ú¾Ù…Ù	https://ww	Ø¨Ø§Ù„Û	entertainm	[15745, 12008, 4807, 12641, 4166, 700, 6391, 5310, 1266, 70, 8870, 70, 13641	152
15	Ú†Ù†Ø¯Ù	https://ww	Ø¨Ø§Ù„Û	entertainm	[15745, 12008, 3592, 6436, 49510, 168288, 507, 211303, 7, 9, 1926, 507, 1252	123
16	â€™Ù¾Ø´	https://ww	ÙÙ„Ù… Ù.	entertainm	[8740, 38590, 1266, 4897, 17323, 15372, 4126, 12751, 951, 105, 83, 6302, 297	111
17	Ø¹Ø±ÙÛŒ	https://ww	Ø¨Ø§Ù„Û	entertainm	[15745, 12008, 1939, 8122, 46651, 158, 6810, 20283, 3159, 8369, 2042, 16654	132
18	ÙÙ„Ú© Ø	https://ww	Ú¯Ù„Ù„ÙˆÚ	entertainm	[32609, 5292, 8027, 8485, 8369, 3581, 700, 105, 83, 211303, 7, 17356, 783, 20	131

Figure 2: Dataset after Tokenizing

Additionally, there was little to no class imbalance in our data as we scrapped multiple sites.

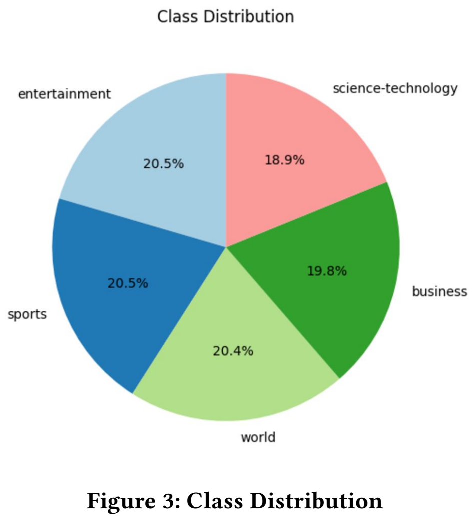


Figure 3: Class Distribution

2.3 Vectorizing our Dataset

For each article, a vector representation was created where each dimension corresponded to a token in the vocabulary. The value at each dimension was the frequency of the token in the article, forming a bag-of-words representation. This method effectively captured the text's structure while maintaining computational simplicity. Additionally, the data was split into training and testing sets, ensuring stratified sampling to balance the class distribution. Labels were mapped to numerical indices, facilitating compatibility with machine learning algorithms.

3 MODELS

3.1 Logistic Regression

Logistic regression is a probabilistic model that predicts the likelihood of an input belonging to each class, making it a natural choice for classification tasks.The process begins with the input features, which in this case are vectors representing the tokenized content of the articles. These vectors are combined linearly with a set of weights and biases. Each weight represents the importance of a specific feature (token) for predicting a particular class, while the biases account for adjustments needed when all input features are zero. The output of this combination is a set of raw scores, one for each class, often referred to as logits. These logits are then passed through the softmax function, which transforms them into probabilities. The softmax ensures that the probabilities for all classes sum to 1, making it suitable for multi-class classification. Each probability indicates how likely the model thinks the input belongs to that class.

The model evaluates its predictions using a loss function (categorical cross-entropy), which measures the difference between predicted probabilities and actual labels. To improve, it uses gradient descent, a process that adjusts weights and biases based on the gradients of the loss function. This iterative process continues over multiple epochs, with the model refining its parameters to minimize the loss.During training, the model iteratively refines its parameters, gradually becoming more accurate in classifying the articles into their respective categories. By the end of training, the model is capable of making reliable predictions based on the patterns it has learned from the data.

Logistic regression was chosen because it is well-suited for multi-class classification tasks like this one. It is simple to implement, computationally efficient, and works effectively with high-dimensional data, such as the word vectors created from tokenized articles. Its ability to provide probabilistic outputs for class predictions also makes it a reliable baseline model for text classification.

3.2 Multinomial Naive Bayes

The Multinomial Naive Bayes (MNB) model was selected as our second model classification task due to its simplicity, computational efficiency, and suitability for text data which aligned with our project. The algorithm is known to perform well with discrete features like word counts or token probabilities, making it a logical choice for our news dataset. Considering all the strengths of Multinomial Naïve Bayes, it was decided to implement it in hopes to achieve an accurate and reliable model. As with the implementation of Logistic Regression, the articles were pre-tokenized into lists of words (tokens) before any processing. The process is applied to both training and testing datasets to create numerical matrices (X_train_vectors and X_test_vectors). This representation allows machine learning models to process text data. The data was split into 70% training and 30% testing data to provide a fair evaluation.

The implementation was done using NumPy and Pandas for data manipulation and Matplotlib/Seaborn for visualization. A custom class (MultinomialNaiveBayes) implementation of Multinomial Naive Bayes was used to implement the algorithm. The fit method calculates the necessary probabilities for classification by computing class priors (log probabilities of each class based on their occurrence in the training data) and word likelihoods (log probabilities of each word given a class, with Laplace smoothing to handle unseen words). These probabilities are derived by analyzing the feature matrix of our dataset, which represents the tokenized dataset, and the target labels. The predict method classifies new samples by calculating a score for each class based on the sum of its prior and the likelihoods of the features, assigning the class with the highest score. This implementation utilizes the probabilistic reasoning to classify new articles with accurate labels.

The model does not require extensive hyperparameter tuning, which allows for faster experimentation. No iterative optimization (for example gradient descent) was necessary since MNB is a probabilistic model rather than a parametric one requiring backpropagation. Hence, the training involved straightforward counting operations. The model was evaluated using Accuracy, Precision, F1 Score, Recall, and Confusion Matrix.

3.3 Fully Connected Neural Network

For our third model a Fully Connected Neural Network (FCNN) was selected for classification due to its ability to model complex, nonlinear relationships between high-dimensional features, which is essential for our purpose. The model architecture is fully connected and comprises three key layers:

- Input Layer: Takes numerical vectors representing tokenized text data as input. These vectors were generated using a bag-of-words approach, where tokens were mapped to their respective indices in the vocabulary and converted into feature vectors.
- Hidden Layers: The network includes two hidden layers, with 512 neurons in the first layer and 256 neurons in the second layer. Each layer uses ReLU (Rectified Linear Unit) activation functions, which introduce non-linearity into the network, allowing it to learn complex features in the data.
- Output Layer: The output layer contains neurons equal to the number of classes in the dataset. It outputs logits, which represent the unnormalized scores for each class. These logits are used for calculating the CrossEntropy loss during training.

The model leverages the Adam optimizer, a widely used optimization algorithm that combines the benefits of momentum and adaptive learning rates. It updates the model weights efficiently to minimize the loss function. The CrossEntropy loss function was chosen for this multi-class classification task as it effectively measures the difference between predicted and actual class probabilities. The input data was preprocessed by converting tokenized content into feature vectors, enabling compatibility with the neural network. The model was trained iteratively for 100 epochs with a learning rate of 0.00001, where the weights were updated using backpropagation and gradient descent.

This architecture allows the neural network to extract and utilize features from the tokenized text data, learning representations that are useful for distinguishing between the different classes in the dataset.

4 RESULTS

4.1 Logistic Regression

The model achieved impressive results, with a best macro-average accuracy of 95% after testing learning rates in the range of 0.01 to 0.1. The optimal performance was observed at a learning rate of 0.05 after 1000 epochs.The macro average precision, recall and F1 scores came out to be 0.95 highlighting fair class distribution. The confusion matrix (Figure below) highlights the classification performance across all categories. Most data points were classified correctly, as evident from the high values along the diagonal indicating that the model effectively differentiated between the categories. These results are reasonable given the quality of preprocessing, feature engineering, and the well-tuned model architecture.

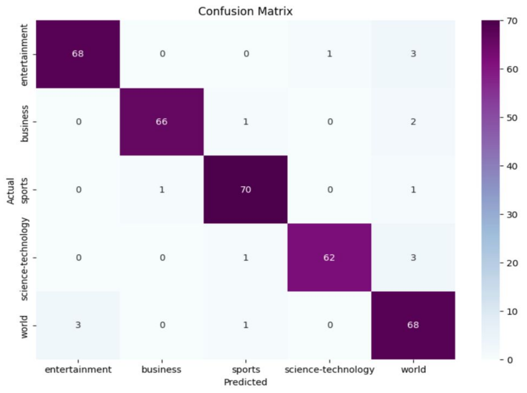


Figure 4: Confusion Matrix

4.2 Multinomial Naive Bayes

Based on the probabilities derived from the analysis of the tokenized news articles, the Multinomial Naive Bayes classifier achieved notable performance metrics. The model effectively predicted the class labels of test data with high accuracy of 96%, showcasing its suitability for text classification tasks. The confusion matrix (figure below) indicated that most predictions aligned with the true labels, with minimal misclassifications. Precision, recall, and F1 scores across classes highlighted a balanced performance with the value of 0.96, demonstrating that the model handled class imbalances well. These results validate the effectiveness of using MNB for processing tokenized textual data.

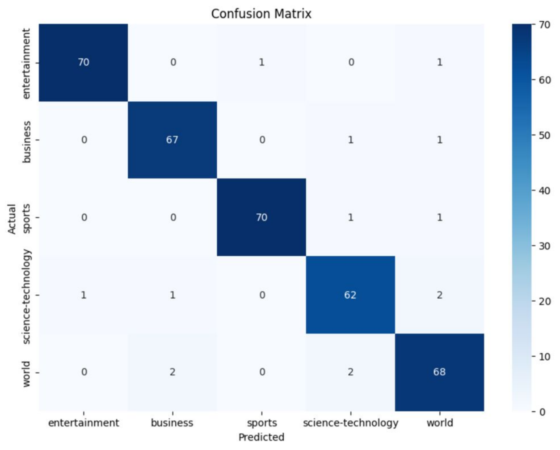


Figure 5: Confusion Matrix

4.3 Neural Network

The Neural Network model delivered outstanding performance, achieving an accuracy of 96.58% with precision, recall, and F1-score metrics averaging 0.97. Through careful optimization of the architecture and hyperparameters, the model effectively extracted meaningful features from the tokenized text data. The confusion matrix demonstrates strong classification capability, with a dominant diagonal indicating that the majority of predictions were accurate. The model excelled across all categories, with only a few misclassifications. For example, in the "entertainment" category it showed minor overlaps with other classes, but these had a negligible effect on overall accuracy. A low test loss of 0.1807 further highlights the model's reliability and its ability to generalize well to unseen data. Moreover, we compared our model with Sklearn's MLPClassifier as well which gave an accuracy of 96.01

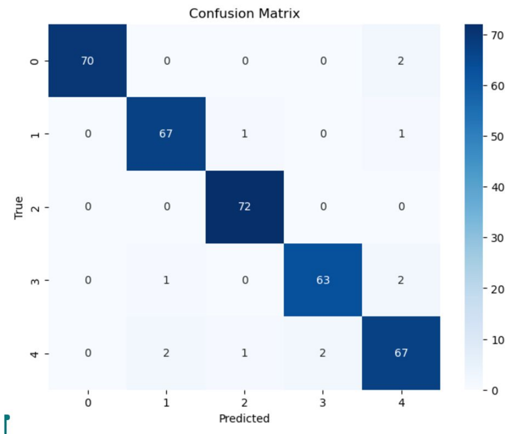


Figure 6: Confusion Matrix

5 CONCLUSION

Three models were implemented for the classification task: Logistic Regression, Multinomial Naive Bayes (MNB), and a Fully Connected Neural Network (FCNN).

- Logistic Regression is a probabilistic model that uses a linear combination of input features, applies the softmax function to compute class probabilities, and optimizes using gradient descent with a CrossEntropy loss. It is simple, efficient, and effective for high-dimensional text data.
- Multinomial Naive Bayes (MNB) is a probabilistic model designed for discrete features like word counts. It calculates class priors and word likelihoods, using probabilistic reasoning to classify samples. It is computationally efficient, doesn't require iterative optimization, and works well for text data.
- Fully Connected Neural Network (FCNN) is a deep learning model with an input layer (bag-of-words representation), two hidden layers (512 and 256 neurons with ReLU activation), and an output layer. It is optimized using the Adam optimizer and trained iteratively with backpropagation. It can model complex, non-linear relationships, making it powerful for high-dimensional data.

The results of the three models demonstrate their effectiveness for text classification, with each offering distinct advantages. Logistic Regression achieved a strong macro-average accuracy of 95%, showcasing balanced performance across classes due to its probabilistic approach and robust preprocessing. Multinomial Naive Bayes slightly outperformed Logistic Regression with an accuracy of 96%, excelling in handling class imbalances and making it a reliable choice for tokenized textual data. However, the Neural Network delivered the best performance, achieving an impressive accuracy of 96.58% and an F1-score of 0.97, thanks to its ability to model complex relationships and extract meaningful features from the data. All three models demonstrated strong performance for the text classification task. Multinomial Naive Bayes and Logistic Regression proved to be efficient and effective, with MNB slightly outperforming Logistic Regression in terms of accuracy. However, the Neural Network excelled, achieving the highest accuracy and F1-score, leveraging its ability to learn complex relationships in the data. While Logistic Regression and MNB are simpler and computationally less demanding, the Neural Network's performance justifies its added complexity, making it the most reliable model for this task.[3]

5.1 Future Considerations

For future work, several experiments could be explored to enhance performance further. Training custom embeddings tailored to the dataset could capture domain-specific semantics more effectively. Testing state-of-the-art models like BERT could provide deeper contextual understanding and potentially improve accuracy. Additionally, evaluating the model on a much larger dataset could test its scalability and robustness, offering insights into its generalization capabilities.

REFERENCES

[1] Sarica S, Luo J. Stopwords in technical language processing. PLoS One. 2021 Aug 5;16(8):e0254937. doi: 10.1371/journal.pone.0254937. Erratum in: PLoS One. 2024 Dec 5;19(12):e0315195. doi: 10.1371/journal.pone.0315195. PMID: 34351911; PMCID: PMC8341615.

[2] hadidev/gpt2-urdu-tokenizer-withgpt2 · Hugging Face. (n.d.). https://huggingface.co/hadidev/gpt2-urdu-tokenizer-withgpt2

[3] Siddiqui, M. I., Aslam, M. (2021). Hierarchical Text Classification of Urdu News using Deep Neural Network. ResearchGate. Retrieved from https://www.researchgate.net/publication/353066845_Hierarchical_Text_Classification_of_Urdu_News_using_Deep_Neural_Network