ML Group 5 Project Report

Mahad Amir*
26100249@lums.edu.pk
LUMS

Uzair Ahmad*
26100324@lums.edu.pk
LUMS

Sarfraz Ahmad*
26100145@lums.edu.pk
LUMS

Saadi Humayun*
26100214@lums.edu.pk
LUMS

Muhammad Daniyal*
26100266@lums.edu.pk
LUMS

Abstract

This project tackles the challenge of organizing unstructured Urdu-language news data to create a personalized news categorization system. Urdu, despite being one of the most widely spoken languages, lacks robust tools for automated content classification and personalized delivery. To address this gap, a dataset was curated through web scraping from prominent Urdu news websites. The collected data was categorized into predefined segments, such as entertainment, business, and sports, using advanced machine learning techniques. Specifically, Support Vector Machines (SVM), Multinomial Naive Bayes (MNB), and Neural Networks (NN) were implemented and evaluated for classification accuracy. The project demonstrates how these models can transform unstructured textual data into organized and accessible information. By streamlining access to relevant news, the proposed system resolves issues related to information overload and language-specific limitations, providing a tailored news experience for Urdu-speaking users.

Keywords

Urdu language, machine learning, Support Vector Machine (SVM), Neural Networks (NN), Multinomial Naive Bayes (MNB), news categorization, personalized news system, natural language processing (NLP), data scraping, digital accessibility.

ACM Reference Format:
Mahad Amir, Uzair Ahmad, Sarfraz Ahmad, Saadi Humayun, and Muhammad Daniyal. 2024. ML Group 5 Project Report. In *Proceedings of (Group 5).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The exponential growth of digital content has necessitated the development of systems capable of automatically categorizing and delivering personalized information. While such systems are commonplace for widely spoken languages like English, resources for Urdu remain limited. This disparity restricts Urdu-speaking audiences from fully utilizing modern information retrieval technologies. To address this challenge, this project aims to build a personalized news categorization system tailored to Urdu, leveraging state-of-the-art machine learning models.

The project begins with data collection, scraping over 3,266 articles from major Urdu news websites such as Hum News, Samaa, and Geo. The articles are categorized into predefined segments, including entertainment, sports, business, world, and science-technology, forming a balanced dataset for training machine learning models. We implemented and compared the performance of Support Vector Machines (SVM), Multinomial Naive Bayes (MNB), and Neural Networks (NN) to identify the most effective model for this task.

By transforming unstructured data into organized information, this system addresses key problems faced by Urdu-speaking users, such as difficulty in accessing relevant content and lack of personalized recommendations. The use of these machine learning techniques not only demonstrates the feasibility of Urdu content classification but also highlights the potential for future developments in personalized digital solutions for underserved languages.

2 Methodology

This section outlines the methodology adopted in this project, detailing the processes of data collection, data cleaning, exploratory data analysis, and feature engineering. These steps were crucial for preparing the dataset for machine learning tasks.

2.1 Data Collection

The initial phase of the project involved gathering a large corpus of Urdu-language news articles. Given the limited availability of pre-existing datasets, web scraping techniques were employed to collect articles directly from various Urdu news websites. The sources included Hum News, Samaa, Express, Geo, City42, Neo News, 24 Urdu, and Dunya News.

One significant challenge encountered was the blocking of automated requests by certain websites, notably Samaa Urdu. To overcome this, we utilized Selenium, a web automation tool that simulates human interaction with websites. This approach enabled us to bypass restrictions and collect articles effectively.

The final dataset comprised articles distributed across multiple news sources, as shown in Table 1.

The articles were further categorized into predefined segments such as entertainment, sports, business, world, and science-technology. This categorization was essential for subsequent machine learning tasks.

Group 5,
,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

Table 1: Article Counts by News Source

News Source	Article Count
Hum News	950
Samaa	840
Com	493
Express	329
Geo	300
City42	112
Neo News	90
24 Urdu	80
Dunya News	72

2.2 Data Cleaning

Data cleaning was a crucial step in ensuring the quality and reliability of the dataset. Several preprocessing procedures were employed:

2.2.1 Handling Missing and Duplicate Values. Rows with missing content were removed to maintain data integrity, and duplicate entries were eliminated to prevent redundancy and bias in the dataset.

2.2.2 Text Standardization. The textual fields, including the `title` and `content` columns, were standardized. Non-Arabic characters, extraneous symbols, and formatting artifacts were removed using regular expressions. This ensured uniformity across the text data, which is critical for natural language processing.

2.2.3 Numerical Conversion. English numerals in the `content` field were converted to Urdu numerals, particularly to maintain the semantic relevance of business-related articles. This step was crucial for accurate numerical interpretation by machine learning models.

2.2.4 Category Encoding. The categorical labels in the `gold_label` field were mapped to numerical values to facilitate machine learning tasks. This transformation allowed the models to process the data effectively.

2.3 Exploratory Data Analysis

Exploratory data analysis (EDA) provided insights into the structure and characteristics of the dataset:

- **Class Distribution Analysis:** The distribution of articles across the predefined categories (entertainment, sports, business, world, and science-technology) was analyzed using a bar chart. The counts of articles were relatively balanced, ranging from 617 to 671 per category.
- **Text Length Analysis:** The length of each article was calculated and stored in a new column, `text_length`. A histogram was plotted to examine the variability in article lengths.
- **Average Article Length by Category:** The average article length was computed for each category and visualized using a bar chart to identify potential distinguishing features.
- **Article Distribution by Source:** The contribution of each news source to the dataset was analyzed by extracting domain names from the article URLs and plotting the article counts per source.

2.4 Feature Engineering

Feature engineering transformed the textual data into formats suitable for machine learning:

2.4.1 Text Vectorization Using TF-IDF. The textual content was converted into numerical feature vectors using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. The number of features was limited to 1,000 to manage computational complexity while retaining essential information.

2.4.2 Dimensionality Reduction with PCA. Principal Component Analysis (PCA) was applied to reduce the high-dimensional TF-IDF feature space to two dimensions. This enabled visualization of the data distribution across categories using a scatter plot, providing insights into its separability.

2.5 Final Dataset Preparation

After completing the data cleaning and feature engineering processes, the dataset consisted of 3,266 articles, balanced across the five categories. The refined dataset was ready for model training, with standardized text, encoded labels, and TF-IDF features ensuring compatibility with machine learning algorithms.

This robust methodology laid a solid foundation for developing accurate and reliable classification models for Urdu news articles, addressing the challenges of unstructured data in underserved languages.

3 Support Vector Machine (SVM)

Support Vector Machine (SVM) was chosen as one of the primary models for this project due to its robustness in handling high-dimensional data and its effectiveness in text classification tasks. SVM is particularly suitable for problems with clear margin separation, making it an excellent choice for our multi-class text categorization problem.

3.0.1 Implementation Details. In our implementation, we used the one-versus-all (OvA) classification strategy to handle the multi-class nature of the dataset. The training phase utilized Sequential Minimal Optimization (SMO), an iterative algorithm designed to optimize the quadratic programming problem that arises in SVM.

Key aspects of the implementation include:

- **Alpha and Bias Optimization:** During training, we continuously updated the Lagrange multiplier (α) parameters and the bias term to minimize the hinge loss function. The process was iteratively performed until the α values reached a low threshold, ensuring convergence and reducing computational complexity.
- **Cross-Validation:** To ensure robust evaluation, k-fold cross-validation was applied during training. This method divided the training data into k subsets, training the model on $k-1$ subsets while validating it on the remaining subset.
- **Stopping Criterion:** The optimization stopped when the change in α values reached a predefined threshold, minimizing unnecessary computation while maintaining performance.

- **Hinge Loss Function:** The SVM loss function penalized misclassified samples, encouraging the model to maximize the margin between classes.

3.0.2 Performance Metrics. The model achieved high accuracy during both cross-validation and testing phases:

- **Cross-Validation Accuracy:** 89.97%
- **Test Set Accuracy:** 86.85%

The performance across individual classes is detailed in Table 2.

Table 2: SVM Performance Metrics by Class

Class	Precision	Recall	F1-Score	Support
0 (Entertainment)	0.77	0.97	0.86	125
1 (Business)	0.88	0.86	0.87	140
2 (Sports)	0.99	0.84	0.91	129
3 (Science-Technology)	0.85	0.84	0.84	128
4 (World)	0.89	0.83	0.86	132

3.0.3 Confusion Matrix and Heatmap. The confusion matrix for the test set illustrates the classification performance, highlighting the correctly and incorrectly classified instances for each category. A heatmap is provided for better visualization (see Figure 1).

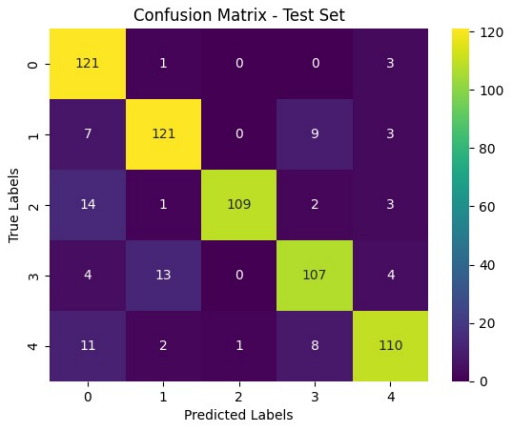

Figure 1: Confusion Matrix Heatmap for SVM

3.0.4 Evaluation and Analysis. The model performed well across all categories, achieving balanced precision, recall, and F1-scores. However, some variations were observed:

- **High Performance in Sports:** The sports category achieved the highest F1-score (0.93), likely due to the distinctive nature of the vocabulary used in this category.
- **Challenges in Science-Technology:** The science-technology category exhibited slightly lower recall (0.84), indicating that some instances were misclassified, potentially due to overlaps in vocabulary with other categories.
- **Balanced Metrics:** The macro average precision, recall, and F1-scores were consistent at 0.88, reflecting balanced performance across categories.

The label mapping used in the model is as follows:

- Entertainment: 0
- Business: 1
- Sports: 2
- Science-Technology: 3
- World: 4

Overall, the SVM demonstrated strong performance in classifying Urdu news articles, making it a viable model for this multi-class categorization task.

4 Neural Network (NN)

Neural Networks (NN) were employed as part of this project due to their ability to learn complex patterns and relationships in data. With their capacity to model non-linear decision boundaries, NNs are well-suited for text classification tasks, particularly in high-dimensional feature spaces such as those generated by TF-IDF vectorization.

4.0.1 Motivation for Using Neural Networks. Neural Networks are known for their adaptability and scalability in handling multi-class classification problems. Unlike traditional models, NNs can capture intricate patterns in data, making them effective in contexts where relationships between features are not linear. For this project, the use of NNs was motivated by the potential for higher accuracy in categorizing Urdu news articles, leveraging the extensive feature set derived from textual content.

4.0.2 Implementation Details. The NN implemented in this project featured a feedforward architecture with the following specifications:

- **Input Layer:** The input layer size matched the feature dimensions, accepting an input vector of size `input_size`.
- **Hidden Layers:**
  - Layer 1: 512 neurons with ReLU activation.
  - Layer 2: 256 neurons with ReLU activation.
  - Layer 3: 128 neurons with ReLU activation.
- **Output Layer:** The output layer consisted of `num_classes` neurons, with a softmax activation function to produce probability scores for each class.

The training setup included:

- **Loss Function:** CrossEntropyLoss, suited for multi-class classification problems.
- **Optimizer:** Adam optimizer with a learning rate of 0.001, ensuring efficient and adaptive gradient updates.
- **Training Epochs:** The model was trained for 40 epochs to ensure convergence.

4.0.3 Strengths of the Model. The NN demonstrated several strengths during implementation and evaluation:

- **Ability to Learn Complex Patterns:** The deep architecture allowed the model to learn non-linear relationships in the data, resulting in high classification accuracy.
- **Adaptability:** The use of ReLU activations and the Adam optimizer facilitated efficient learning across layers, preventing vanishing gradient problems.

- **Robust Multi-Class Performance:** The softmax output layer ensured that the model effectively handled the multi-class nature of the problem.

4.0.4 Performance Analysis and Findings. The Neural Network achieved a final accuracy of 94% on the test set. Detailed performance metrics are presented in Table 3.

Table 3: Neural Network Performance Metrics by Class

Class	Precision	Recall	F1-Score	Support
0 (Entertainment)	0.96	0.98	0.97	125
1 (Business)	0.94	0.96	0.95	140
2 (Sports)	0.98	0.96	0.97	129
3 (Science-Technology)	0.92	0.91	0.91	128
4 (World)	0.91	0.92	0.91	132

The confusion matrix heatmap for the test set is shown in Figure 2, providing a visual representation of the model's classification performance.

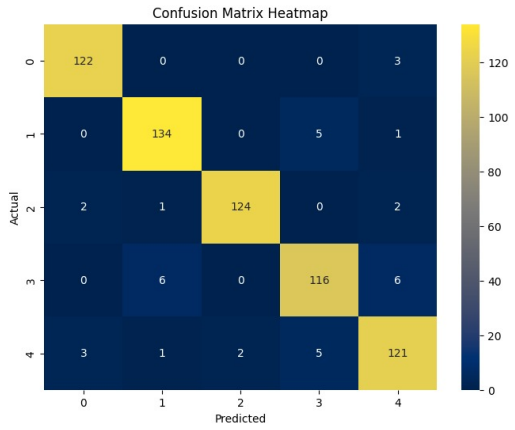

Figure 2: Confusion Matrix Heatmap for Neural Network

Key findings include:

- **High Precision in Sports Category:** The sports category achieved the highest precision (0.98), likely due to distinct vocabulary and content structure.
- **Balanced Macro Average:** The macro average for precision, recall, and F1-score was consistent at 0.93, indicating balanced performance across all classes.
- **Room for Improvement in Science-Technology:** While the model performed well overall, the science-technology category showed slightly lower recall (0.88), possibly due to overlaps in features with other categories.

4.0.5 Limitations of the Model. Despite its strong performance, the Neural Network had some limitations:

- **Computational Complexity:** The deep architecture required significant computational resources, particularly during training.
- **Overfitting Risk:** With a high number of parameters, there was a potential risk of overfitting to the training data. Regularization techniques could be explored further to mitigate this.
- **Sensitivity to Hyperparameters:** The performance was highly sensitive to hyperparameter settings, requiring careful tuning to achieve optimal results.

Overall, the Neural Network demonstrated excellent accuracy and robustness, establishing its suitability for the task of Urdu news article classification. Its ability to generalize across diverse categories highlights the potential of deep learning in natural language processing for underserved languages.

5 Multinomial Naive Bayes (MNB)

Multinomial Naive Bayes (MNB) was selected as one of the models for this project due to its effectiveness in text classification tasks, particularly when features are derived from word frequencies. MNB is well-suited for discrete data and works efficiently in scenarios where text data is represented as a Bag of Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF) matrix.

5.0.1 Motivation for Using Multinomial Naive Bayes. The Multinomial Naive Bayes model leverages the probabilistic relationships between word frequencies and classes, making it highly effective for tasks where specific words or phrases frequently correlate with particular categories. Its computational efficiency and simplicity made it an ideal choice for this project, especially given the repetitive nature of domain-specific keywords in Urdu news articles.

5.0.2 Implementation Details. The implementation of MNB involved several key steps, tailored to the unique requirements of the dataset:

1. Data Preprocessing. Preprocessing was crucial to ensure the data was clean and well-structured. The following steps were performed:

- **Duplicates Removal:** Duplicate rows were removed based on the `content` and `title` columns to eliminate redundancy.
- **Handling Missing Data:** Rows with missing values were dropped to maintain data integrity.
- **Feature Enhancement:** The `title` was appended twice to the `content` to emphasize its importance, as the title often contains concise, category-relevant keywords.
- **Data Splitting:** The data was split into training and testing sets using `train_test_split` with a random state of 42 to ensure reproducibility.

2. Bag of Words Implementation. The textual data was vectorized using a custom `BagOfWords` implementation:

- A vocabulary was built, assigning an index to each unique token in the corpus.
- Tokens appearing only once were filtered out to reduce noise.
- Input text was converted into feature vectors based on token frequencies, providing discrete input features for the MNB model.

3. Multinomial Naive Bayes Model. The Multinomial Naive Bayes model was implemented with the following considerations:

- **Training:**
  - Calculated class priors, representing the probability of each class in the training set.
  - Estimated conditional probabilities for each word given a class using Laplace smoothing to handle unseen tokens.
  - Computed log-probabilities to prevent numerical underflow during multiplication of small probabilities.
- **Prediction:** For each test sample, the model calculated the likelihood for each class based on word frequencies and selected the class with the highest likelihood.

5.0.3 Strengths of the Model. The Multinomial Naive Bayes model exhibited the following strengths:

- **Efficiency:** The model was computationally efficient, making it well-suited for large datasets.
- **Domain-Specific Keyword Repetition:** Categories like Business and Entertainment contained repetitive keywords (e.g., "stocks," "gold prices," or "Bollywood actors"), which the model effectively leveraged.
- **Robust Handling of Sparse Data:** The model performed well with sparse data generated by the Bag of Words representation.

5.0.4 Performance Analysis and Findings. The Multinomial Naive Bayes model achieved a test accuracy of 93.51%. Detailed performance metrics are provided in Table 4.

Table 4: Multinomial Naive Bayes Performance Metrics by Class

Class	Precision	Recall	F1-Score	Support
0 (Entertainment)	0.94	0.94	0.94	156
1 (Business)	0.97	0.96	0.97	162
2 (Sports)	0.94	0.87	0.90	158
3 (Science-Technology)	0.97	0.95	0.96	160
4 (World)	0.86	0.95	0.90	165

The confusion matrix heatmap is shown in Figure 3, providing a visual representation of classification performance.

Key findings include:

- **High Accuracy:** The model achieved an overall test accuracy of 93.51%.
- **Strong Performance in Business Category:** The Business category had the highest precision (0.97), reflecting the model's ability to capitalize on repetitive domain-specific keywords.
- **Balanced Metrics:** The macro average for precision, recall, and F1-score was consistent at 0.94, indicating reliable performance across all categories.

5.0.5 Limitations of the Model. Despite its strong performance, the Multinomial Naive Bayes model had some limitations:

- **Feature Independence Assumption:** The model assumes that features (words) are independent, which may not hold true in real-world text data where word dependencies exist.
- **Vocabulary Sensitivity:** The model's performance heavily relies on the quality and completeness of the vocabulary generated by the Bag of Words representation.

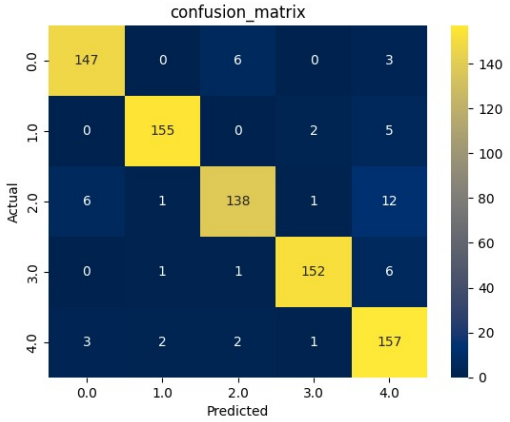

Figure 3: Confusion Matrix Heatmap for Multinomial Naive Bayes

- **Difficulty with Rare Features:** Words that appear infrequently across categories may not significantly influence predictions due to their low impact on likelihood calculations.

Overall, the Multinomial Naive Bayes model demonstrated exceptional accuracy and efficiency, making it a reliable option for text classification tasks, particularly in scenarios where domain-specific keywords are prevalent.

6 Comparative Analysis of Models

In this section, we present a detailed comparative analysis of the three models implemented in this project: Support Vector Machine (SVM), Neural Network (NN), and Multinomial Naive Bayes (MNB). The analysis is based on their performance metrics, strengths, and limitations, with a focus on identifying the model that best suits the task of Urdu news categorization.

6.1 Performance Comparison

The performance of each model is evaluated using metrics such as accuracy, precision, recall, and F1-score, as well as their ability to handle the specific challenges of the dataset. Table 5 summarizes the performance metrics for each model.

Table 5: Performance Metrics Comparison of SVM, NN, and MNB

Metric	SVM	NN	MNB
Accuracy (%)	87.92	93.00	93.51
Macro Precision (%)	88.00	93.00	94.00
Macro Recall (%)	88.00	93.00	93.00
Macro F1-Score (%)	88.00	93.00	94.00

6.2 Model Strengths and Weaknesses

Each model demonstrated unique strengths and limitations during training and evaluation, as detailed below:

6.2.1 Support Vector Machine (SVM). **Strengths:**

- Robust to high-dimensional data, leveraging the hinge loss function to optimize the hyperplane that separates classes.
- Effective in handling moderately imbalanced datasets due to its margin maximization principle.

Weaknesses:

- Computationally expensive, particularly during the training phase, due to the need to solve quadratic programming problems iteratively.
- Lower accuracy (87.92%) compared to other models, indicating difficulty in capturing nuanced relationships in the text data.

6.2.2 Neural Network (NN). **Strengths:**

- Achieved high accuracy (93.00%) by learning complex, non-linear relationships in the data.
- Demonstrated balanced performance across all categories, with macro metrics at 93%.
- Scalability and adaptability to additional data or features.

Weaknesses:

- Computationally intensive, requiring significant resources for training and inference.
- Sensitive to hyperparameter tuning and prone to overfitting if not properly regularized.

6.2.3 Multinomial Naive Bayes (MNB). **Strengths:**

- Computationally efficient, with fast training and prediction times.
- Achieved the highest accuracy (93.51%), outperforming both SVM and NN in this dataset.
- Particularly effective for domain-specific text categorization, leveraging repetitive keywords.

Weaknesses:

- Relies on the assumption of feature independence, which may not hold in real-world text data.
- Vocabulary-sensitive, with performance heavily dependent on the quality and completeness of the feature set.

6.3 Analysis and Insights

The comparative analysis reveals the following key insights:

- **MNB Outperformed in Accuracy:** The Multinomial Naive Bayes model achieved the highest accuracy (93.51%), largely due to its ability to leverage repetitive domain-specific keywords effectively.
- **NN Captured Complexity:** The Neural Network demonstrated its strength in capturing non-linear relationships, achieving balanced metrics across all categories. However, its computational intensity and sensitivity to hyperparameters are notable drawbacks.
- **SVM for Simplicity:** While the Support Vector Machine achieved slightly lower accuracy (87.92%), it remains a viable option for smaller datasets or scenarios where computational resources are limited.
- **Suitability to Dataset Characteristics:** The dataset's domain-specific keyword repetition favored MNB, while the neural network's ability to generalize performed well with the diverse and balanced dataset.

6.4 Conclusion of Comparative Analysis

The Multinomial Naive Bayes model is the most effective for this specific task, given its high accuracy and computational efficiency. However, the Neural Network's adaptability and scalability make it a strong candidate for future extensions or datasets with more complex relationships. The Support Vector Machine, while outperformed in this case, provides a robust baseline and may excel in scenarios with different data characteristics or constraints.

7 Conclusion

The Multinomial Naive Bayes (MNB) and Neural Network (NN) models outperformed Support Vector Machine (SVM) due to their ability to handle the complexities of the dataset. MNB excelled with an accuracy of 93.51%, leveraging its probabilistic approach and the repetitive nature of domain-specific keywords, which aligned well with its Bag of Words representation. NN achieved a close 93.00% accuracy by learning complex, non-linear relationships, demonstrating its adaptability and robustness in handling balanced and diverse data. In contrast, SVM lagged behind with 87.92% accuracy due to its reliance on linear separability, which proved insufficient for the overlapping feature distributions in the dataset. The results highlight the importance of choosing models suited to the dataset's characteristics, with MNB and NN standing out as the most effective for Urdu news categorization.