Reddit Post Title Classification

Instructor: *Dr.Cumali Türkmenoğlu*

Learning from Data – Final Project

Group Members:

Mukhammad Shaban-2221251368

Abdulrahman Aljabahji-2221251353

Kenan Alemam-2521051363

1. Introduction

1.1 Problem Motivation

Online platforms such as Reddit generate a large volume of user-created content covering a wide range of topics. As this content continues to grow, manually organizing and categorizing posts becomes inefficient and impractical. Automatic text classification systems play an important role in improving content organization, search, recommendation systems, and overall user experience.

Text classification is a core task in machine learning and natural language processing (NLP). It involves assigning predefined categories to textual data based on its content. Although Reddit post titles are short, they often contain meaningful contextual information that can be used to identify their underlying topic. This makes them a suitable and realistic example for studying multi-class text classification using supervised learning techniques.

The motivation of this project is to investigate how different machine learning and deep learning models perform when applied to short-text classification problems and to compare their effectiveness on real-world data.

1.2 Dataset Description

The dataset used in this project consists of Reddit post titles collected through web scraping. Each data instance contains a post title and an associated topic label. The dataset includes a total of **4,518** samples distributed across five categories: technology, science, sports, books, and fitness.

The dataset is relatively balanced, with no category dominating the others, which allows for fair model evaluation. The text contains no missing values, and the average title length is approximately 15 words, making the dataset well-suited for short-text classification tasks. Each sample consists of two attributes: the text of the Reddit post title and its corresponding category label.

2. Data Collection & Preprocessing

2.1 Scraping Methodology

The dataset used in this project was collected from Reddit, a popular online discussion platform where users post content across a wide range of topics. Reddit was chosen due to its diverse subject coverage and the availability of topic-focused communities. Data collection was performed through web scraping, following standard ethical considerations and platform usage policies.

Only post titles were collected, as they provide concise textual information that reflects the main topic of each post. During the scraping process, titles were extracted along with their corresponding category labels based on the subreddit or predefined topic mapping. The scraping process ensured that each collected sample was meaningful, readable, and relevant to one of the selected categories: technology, science, sports, books, and fitness.

After collection, the raw data was stored in a structured format and reviewed to ensure consistency and correctness before further processing. Duplicate entries and incomplete records were removed during this stage.

2.2 Data Statistics and Visualization

After data collection and initial cleaning, the final dataset contained **4,518** Reddit post titles labeled across five categories. The dataset showed a relatively balanced distribution among the classes, which is desirable for supervised classification tasks as it reduces bias toward any single category.

Basic exploratory data analysis was conducted to better understand the dataset. Class distribution analysis showed that each category contained a comparable number of samples, with no extreme imbalance. Additionally, text length analysis revealed that Reddit post titles are generally short, with an average length of approximately **15 words** per title. This confirms that the task is a short-text classification problem.

Visualizations such as bar charts were used to illustrate the distribution of categories, and descriptive statistics were computed to summarize text length characteristics. These

analyses provided useful insights that guided later decisions regarding feature engineering and model selection.

2.3 Preprocessing Pipeline

Before training machine learning models, the text data was preprocessed to improve model performance and reduce noise. Since machine learning algorithms cannot operate directly on raw text, preprocessing is a crucial step in the pipeline.

The preprocessing steps applied in this project include:

- Converting all text to lowercase to ensure consistency.
- Removing non-alphabetic characters and punctuation.
- Tokenizing text into individual words.
- Removing common stopwords that do not contribute meaningful information for classification.

Feature extraction was performed using **Bag-of-Words (BoW)** and **TF-IDF** representations, which transform textual data into numerical feature vectors. TF-IDF was used as the primary representation due to its ability to capture the importance of words across documents.

In addition to text-based features, custom domain-specific features were extracted, including title length, number of digits, and the presence of question or exclamation marks. These features help capture stylistic patterns that may differ across categories.

2.4 Challenges Encountered

Several challenges were encountered during data collection and preprocessing. One challenge was ensuring that scraped data was clean and correctly labeled, as user-generated content often contains informal language and inconsistent formatting. This required careful filtering and validation of collected samples.

3. Methodology

This section describes the feature engineering techniques, machine learning models, training strategy, and hyperparameter tuning process used to solve the multi-class text classification problem.

3.1 Feature Engineering Approaches

Since machine learning models require numerical input, feature engineering plays a critical role in transforming raw text into a suitable representation. In this project, several feature engineering techniques were applied to capture both textual content and stylistic characteristics of Reddit post titles.

3.1.1 Bag-of-Words (BoW)

The Bag-of-Words approach represents text by counting the frequency of each word in a document, ignoring word order. Although simple, BoW provides a strong baseline for text classification tasks. Stopwords were removed, and the maximum number of features was limited to reduce dimensionality and prevent overfitting. BoW was implemented primarily to satisfy feature engineering requirements and to serve as a comparative baseline.

3.1.2 TF-IDF Representation

The primary feature representation used in this project is **Term Frequency–Inverse Document Frequency (TF-IDF)**. Unlike BoW, TF-IDF assigns higher weights to words that are more informative and less frequent across documents. This helps reduce the impact of common words while emphasizing discriminative terms.

TF-IDF is particularly effective for short-text classification tasks such as Reddit post titles, as it produces sparse, high-dimensional feature vectors that work well with linear classifiers. Stopword removal and lowercasing were applied during vectorization, and the feature space was limited to a fixed number of dimensions to improve computational efficiency.

3.1.3 Custom Domain-Based Features

In addition to text-based features, a set of custom, domain-specific features was extracted to capture stylistic patterns that may differ across categories. These features include:

- Title length (number of words)
- Number of digits in the title
- Presence of question marks
- Presence of exclamation marks

These handcrafted features were motivated by domain knowledge, as certain categories such as sports or fitness often include numerical scores or expressive punctuation. Incorporating these features complements textual representations and enhances model interpretability.

3.2 Algorithm Descriptions and Justifications

To ensure a comprehensive comparison, both traditional machine learning models and a deep learning model were implemented.

3.2.1 Logistic Regression

Logistic Regression was used as a baseline linear classifier. Despite its simplicity, it performs well on high-dimensional sparse data such as TF-IDF features. Logistic Regression provides interpretable results and serves as a strong reference point for comparing more complex models.

3.2.2 Multinomial Naive Bayes

Multinomial Naive Bayes is a probabilistic model commonly used in text classification. It assumes conditional independence between words and models word frequencies within each class. Due to its efficiency and strong performance on short texts, Naive Bayes is a standard NLP baseline and was included to evaluate probabilistic modeling approaches.

3.2.3 Linear Support Vector Machine (SVM)

Linear SVM is one of the most effective models for text classification tasks. It attempts to find an optimal separating hyperplane between classes in a high-dimensional feature space. Linear SVM was chosen due to its strong generalization performance, robustness to overfitting, and effectiveness with TF-IDF features.

3.2.4 Multi-Layer Perceptron (MLP)

To satisfy the deep learning requirement, a Multi-Layer Perceptron (MLP) was implemented. The MLP introduces non-linear decision boundaries, allowing the model to capture more complex patterns in the data. While neural networks are not always superior for text classification, the MLP provided an opportunity to compare deep learning with traditional approaches under the same feature representation.

3.3 Hyperparameter Tuning Process

Hyperparameter tuning was conducted to improve model performance and ensure fair comparison. For Linear SVM, the regularization parameter **C** was tuned using grid search combined with 5-fold cross-validation. Several values of C were evaluated, and the optimal value was selected based on cross-validation accuracy.

This process helped balance the tradeoff between model complexity and generalization. Cross-validation ensured that the selected hyperparameters performed consistently across different data splits rather than relying on a single train-test partition.

3.4 Training Strategy

The dataset was split into training and testing sets using an 80/20 ratio, with stratification applied to preserve class distribution across splits. This ensured that each category was adequately represented during both training and evaluation.

To further assess model stability and generalization, 5-fold cross-validation was applied to the best-performing traditional model. Evaluation was conducted using multiple metrics, including accuracy, precision, recall, and F1-score. Confusion matrices and learning curves were also generated to analyze classification behavior and bias–variance characteristics.

4. Results & Analysis

This section presents the experimental results of the implemented models and provides a detailed analysis of their performance. Multiple evaluation metrics, visualizations, and comparative analyses are used to assess the effectiveness of each approach.

4.1 Model Performance Comparison

To evaluate the effectiveness of different classification approaches, four models were trained and tested using the same dataset and feature representation. Table 1 summarizes the classification accuracy achieved by each model on the test set.

Table 1: Model Performance Comparison

Model	Accuracy
Logistic Regression	0.887
Multinomial Naive Bayes	0.884
Linear SVM	0.903
MLP (Neural Network)	**0.908**

The results indicate that all models perform well on the multi-class text classification task. Traditional linear models such as Logistic Regression and Linear SVM demonstrate strong performance, highlighting the effectiveness of TF-IDF features for text data. The Multinomial Naive Bayes model performs competitively despite its simplifying assumptions, serving as a strong probabilistic baseline.

The Multi-Layer Perceptron achieves the highest accuracy, suggesting that introducing non-linear decision boundaries provides a marginal improvement over linear classifiers. However, the performance gap between MLP and Linear SVM is relatively small, indicating that simpler models can still be highly effective for short-text classification.

4.2 Learning Curves and Model Behavior

Learning curves were generated for the Linear SVM model to analyze how performance changes as the size of the training dataset increases. The learning curves plot training accuracy and validation accuracy against different training set sizes.

The training accuracy remains consistently high across all training sizes, indicating that the model is capable of fitting the training data effectively. In contrast, validation accuracy increases steadily as more training data is used and eventually converges toward the training accuracy.

This behavior suggests a good bias–variance tradeoff. The small gap between training and validation accuracy indicates that the model does not suffer from severe overfitting, while the steady improvement in validation performance suggests that the model benefits from additional training data. These observations confirm that the selected feature representation and model complexity are well-suited for the task.

4.3 Error Analysis

To better understand model behavior beyond aggregate metrics, a detailed error analysis was conducted using confusion matrices and class-wise performance metrics. The confusion matrix for the Linear SVM model shows that most predictions lie along the diagonal, indicating high classification accuracy across all categories.

The majority of misclassifications occur between the **science** and **technology** categories. This confusion is expected, as many Reddit post titles discuss topics such as artificial intelligence, data science, or technological research that overlap conceptually with scientific content. For example, titles related to AI research or computational methods may be difficult to assign to a single category without additional context.

The **books** category exhibits slightly lower recall compared to other classes. This can be attributed to the more general language used in book-related titles, which may lack distinctive keywords and resemble content from other categories. In contrast, categories such as **fitness** and **sports** achieve consistently high precision and recall, likely due to their more distinctive vocabulary, including terms related to workouts, scores, and competitions.

This error analysis demonstrates that most misclassifications are semantically reasonable and reflect natural overlap between categories rather than model failure.

4.4 Statistical Significance and Stability Analysis

To assess the stability and reliability of the results, 5-fold cross-validation was applied to the Linear SVM model. Cross-validation results showed consistent accuracy across folds, with only minor variations. This indicates that the observed performance is not dependent on a specific train-test split and that the model generalizes well to unseen data.

Although formal statistical significance tests were not conducted across all models, the small performance differences between models suggest that improvements are incremental rather than drastic. The consistency between cross-validation accuracy and test accuracy further supports the robustness of the experimental results.

4.5 Discussion of Results

Overall, the experimental results demonstrate that both traditional machine learning and deep learning approaches are effective for multi-class text classification of short texts. Linear models perform strongly due to the high-dimensional, sparse nature of TF-IDF features, while the MLP benefits from its ability to model non-linear relationships.

The relatively small performance gap between models suggests that careful feature engineering and proper evaluation strategies are as important as model complexity. These findings highlight the value of systematic experimentation and analysis when selecting machine learning models for real-world text classification tasks.

5. Discussion

This section interprets the experimental results, analyzes model behavior, discusses limitations, and reflects on key lessons learned throughout the project.

5.1 Interpretation of Results

The experimental results demonstrate that all evaluated models are capable of effectively classifying Reddit post titles into multiple categories. Traditional machine learning models, particularly Logistic Regression and Linear SVM, achieved strong performance when combined with TF-IDF feature representations. This confirms that linear classifiers are well-suited for short-text classification tasks with high-dimensional, sparse feature spaces.

Among all models, the Multi-Layer Perceptron (MLP) achieved the highest classification accuracy. This suggests that incorporating non-linear decision boundaries can provide a modest performance improvement over linear approaches. However, the improvement margin was relatively small, indicating that feature representation and data quality play a more critical role than model complexity in this task.

The strong performance of Multinomial Naive Bayes further highlights that even simple probabilistic models can be competitive for text classification when appropriate preprocessing and feature engineering are applied.

5.2 Bias–Variance Analysis

Bias–variance behavior was analyzed using learning curves and cross-validation results. The learning curves for the Linear SVM model showed that training accuracy remained high while validation accuracy steadily increased and converged toward the training performance as more data was used. This indicates low bias and controlled variance.

The small gap between training and validation accuracy suggests that the models generalize well to unseen data and do not suffer from severe overfitting. Cross-validation results further supported this observation by demonstrating consistent performance across different data

splits. Regularization techniques and feature dimensionality control played an important role in maintaining this balance between bias and variance.

Overall, the selected models achieved a favorable bias–variance tradeoff, making them reliable for real-world text classification scenarios.

5.3 Limitations and Future Work

Despite the strong results, this project has several limitations. First, the dataset is limited to Reddit post titles, which are short and may lack sufficient context for accurate classification in ambiguous cases. Including post content or comments could potentially improve classification performance.

Second, the feature representations used in this project are primarily based on word frequency statistics. While TF-IDF is effective, it does not capture semantic relationships between words. Future work could explore word embeddings such as Word2Vec, GloVe, or transformer-based representations to capture deeper contextual meaning.

Additionally, only a basic neural network architecture was explored. Future studies could investigate more advanced deep learning models, such as convolutional or transformer-based architectures, to further improve performance. Expanding the dataset and experimenting with multilingual data could also enhance model robustness.

5.4 Lessons Learned

This project provided practical experience in applying machine learning techniques to real-world text data. One key lesson is that effective feature engineering and proper preprocessing can have a greater impact on model performance than model complexity alone. Linear models combined with TF-IDF features proved to be highly effective for short-text classification tasks.

Another important lesson was the value of systematic evaluation. Techniques such as cross-validation, learning curves, and error analysis helped in understanding model behavior and ensuring reliable performance. Finally, the project highlighted the importance of