



Reddit Post Title Classification

Instructor: Dr.Cumali Türkmenoğlu

**Learning from Data – Final Project Group
Members:**

Kenan Alemam-2521051363

Mukhammad Shaban-2221251368

Abdulrahman Aljabahji-2221251353

THE LINK OF THE GITHUB : [Kinan404/reddit-text-classification: Multi-class classification of Reddit post titles using machine learning](https://github.com/Kinan404/reddit-text-classification)

1. Introduction

1.1 Problem Motivation

Online platforms generate an enormous amount of textual data every day. Reddit, in particular, hosts millions of posts covering a wide range of topics such as technology, science, sports, books, and fitness. As the volume of content grows, manually organizing and categorizing posts becomes inefficient, time-consuming, and impractical. Automatic text classification systems are therefore essential for structuring content, improving search and recommendation systems, and enhancing overall user experience.

Text classification is a fundamental problem in machine learning and natural language processing (NLP). It involves assigning predefined labels to text based on its content. Despite their short length, Reddit post titles often contain strong contextual signals that indicate the main topic of the post. This makes them a suitable and realistic example for studying supervised multi-class text classification. By developing an automated classifier, large-scale text data can be processed efficiently with minimal human intervention.

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 a corresponding topic label. The dataset includes a total of **4,518** samples distributed across five categories: **technology, science, sports, books, and fitness**. The class distribution is relatively balanced, which helps reduce bias during model training and evaluation.

Each record in the dataset consists of two attributes:

- **Text:** the Reddit post title
- **Label:** the category associated with the title

The dataset contains no missing values and represents real-world, user-generated text, which may include informal language and varying writing styles. The average length of a post title is approximately **15 words**, making the task a short-text classification problem.

1.3 Project Objectives

The primary objective of this project is to design and evaluate a supervised machine learning system capable of accurately classifying Reddit post titles into one of five predefined categories. To achieve this goal, the project pursues the following objectives:

1. To preprocess and represent textual data using effective feature engineering techniques.
2. To implement and compare multiple traditional machine learning models and a deep learning model for multi-class classification.
3. To evaluate model performance using standard classification metrics such as accuracy, precision, recall, and F1-score.

2. Data Collection & Preprocessing

2.1 Scraping Methodology

The dataset used in this project was collected from Reddit using web scraping techniques. Reddit was selected due to its wide range of topic-specific communities and the availability of large amounts of publicly accessible textual data. Only post titles were collected, as they provide concise information that reflects the main topic of each post.

During scraping, post titles were extracted along with their corresponding category labels based on predefined topic mappings. Basic filtering was applied to remove duplicate entries and incomplete records. The collected data was then stored in a structured format and reviewed to ensure correctness before further processing.

2.2 Data Statistics and Visualization

After data collection and cleaning, the final dataset contained **4,518** Reddit post titles categorized into five classes: technology, science, sports, books, and fitness. The class distribution was relatively balanced, with each category containing a comparable number of samples. This balanced distribution helps prevent model bias toward any single class.

Exploratory data analysis was conducted to understand the dataset characteristics. Summary statistics showed that post titles are short, with an average length of approximately **15 words**. Basic visualizations, such as class distribution bar charts, were used to verify label balance and to gain insight into the structure of the dataset.

2.3 Preprocessing Pipeline

Before training the models, text preprocessing was applied to prepare the raw data for feature extraction. Preprocessing was handled implicitly through the TF-IDF vectorization process. This included converting text to lowercase, tokenizing titles into words, removing common stopwords, and eliminating non-alphabetic characters.

Feature extraction was performed using Bag-of-Words and TF-IDF representations, with TF-IDF used as the primary feature set due to its effectiveness in highlighting informative terms. In addition to text-based features, simple domain-specific features such as title length and the presence of digits or punctuation were included to capture stylistic patterns.

2.4 Challenges Encountered

One of the main challenges encountered was the short length of Reddit post titles, which limits contextual information and can lead to ambiguity between certain categories, particularly science and technology. Another challenge involved handling noisy user-generated text, which required careful preprocessing to remove irrelevant information while preserving meaningful content.

3. Methodology

This section describes the feature engineering techniques, classification algorithms, hyperparameter tuning process, and training strategy used in this project.

3.1 Feature Engineering Approaches

Machine learning models require numerical representations of text. Therefore, feature engineering was applied to transform Reddit post titles into meaningful numerical features.

The **Bag-of-Words (BoW)** approach was used as a basic text representation by counting word occurrences in each title. While simple, BoW provides a useful baseline for text classification.

The primary feature representation used in this project is **TF-IDF (Term Frequency–Inverse Document Frequency)**. TF-IDF assigns higher weights to words that are important within a specific document and lower weights to common words across the dataset. This representation is particularly effective for short-text classification and works well with linear classifiers.

In addition to textual features, a set of **custom domain-based features** was included to capture stylistic information. These features include title length, number of digits, and the presence of question or exclamation marks. Such features help distinguish categories that exhibit different writing patterns.

3.2 Algorithm Descriptions and Justifications

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

Logistic Regression was used as a baseline linear classifier due to its simplicity, efficiency, and strong performance on high-dimensional sparse data such as TF-IDF features.

Multinomial Naive Bayes is a probabilistic model commonly used in text classification. It models word frequencies within each class and is computationally efficient, making it suitable for short-text data.

Linear Support Vector Machine (SVM) was selected because it is one of the most effective models for text classification tasks. Linear SVM performs well on sparse feature spaces and provides strong generalization performance.

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 and enabling a comparison between traditional machine learning and neural network-based approaches.

3.3 Hyperparameter Tuning Process

Hyperparameter tuning was performed to improve model performance and ensure fair comparison. For the Linear SVM model, the regularization parameter **C** was tuned using grid search combined with **5-fold cross-validation**. Multiple values of C were evaluated, and the value that produced the highest cross-validation accuracy was selected.

This tuning process helped balance model complexity and generalization, reducing the risk of overfitting while maintaining strong predictive performance.

3.4 Training Strategy

The dataset was divided into training and testing sets using an **80/20 split**, with stratification applied to preserve the class distribution across both sets. Models were trained on the training data and evaluated on the unseen test set.

To further assess model stability, **5-fold cross-validation** was applied to selected models. Performance was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. This training strategy ensured reliable evaluation and allowed meaningful comparison between different classification approaches.

4. Results & Analysis

This section presents the experimental results of the implemented models and provides an analysis of their performance using quantitative metrics, visualizations, and error analysis.

4.1 Model Performance Comparison

To evaluate the effectiveness of different approaches, four classification models were trained and tested using the same dataset and feature representations. 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
Multi-Layer Perceptron (MLP)	0.908

The results show that all models perform well on the multi-class classification task. Traditional machine learning models achieve strong performance, particularly Linear SVM, which benefits from the high-dimensional sparse nature

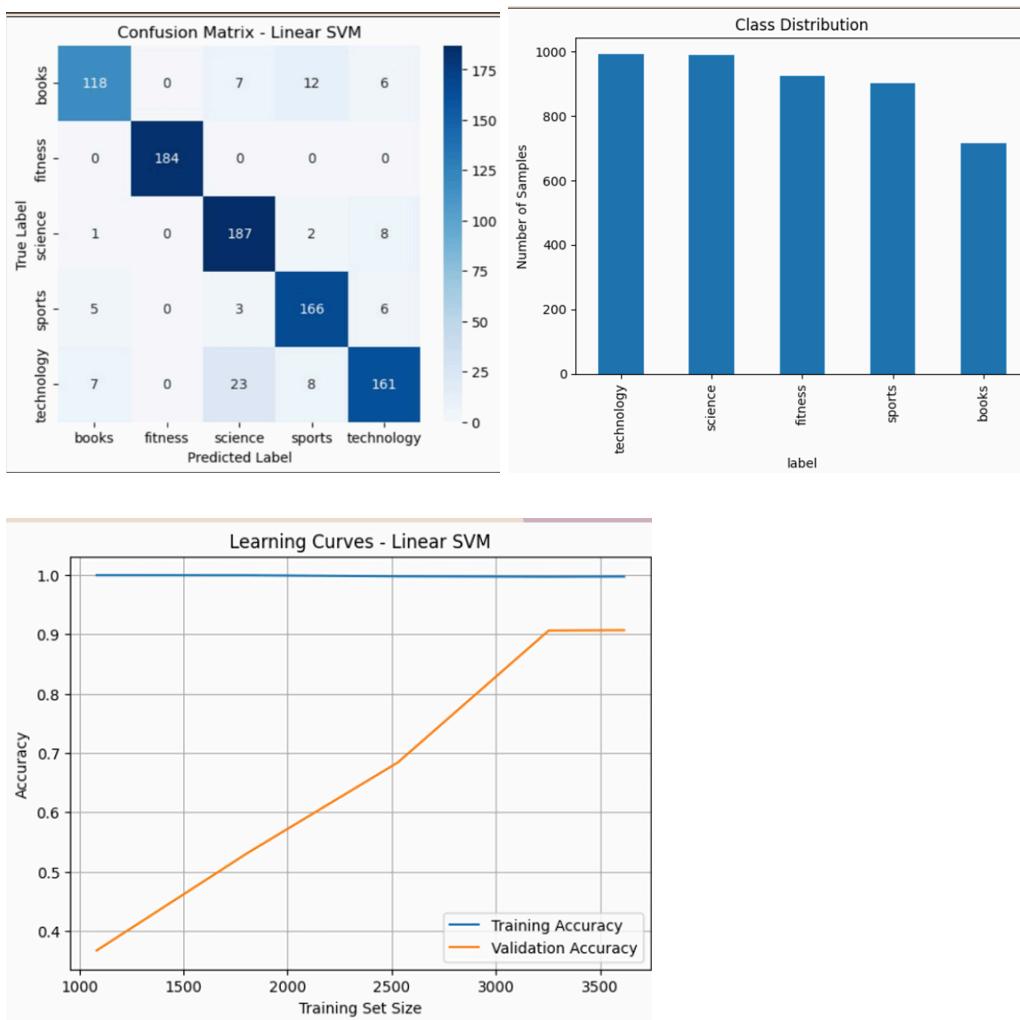
of TF-IDF features. The MLP achieves the highest accuracy, indicating that non-linear models can provide a modest improvement over linear classifiers. However, the difference between the best-performing models is relatively small, suggesting that feature representation plays a crucial role in overall performance.

4.2 Learning Curves and Model Behavior

Learning curves were generated to analyze how model performance changes with increasing training data size. These curves compare training accuracy and validation accuracy across different proportions of the training set.

The learning curves indicate that training accuracy remains consistently high, while validation accuracy gradually increases and converges toward the training performance. This behavior suggests that the models are able to learn meaningful patterns without severe overfitting. The small gap between training and validation accuracy indicates a balanced bias–variance tradeoff and good generalization to unseen data.

These observations confirm that the selected models and feature engineering techniques are appropriate for the dataset and that additional training data could further improve performance.



4.3 Error Analysis

To better understand model behavior beyond aggregate accuracy, error analysis was conducted using confusion matrices and class-wise performance metrics. Most predictions are correctly classified, as indicated by strong diagonal values in the confusion matrices.

The most common misclassifications occur between the **science** and **technology** categories. This confusion is expected, as many Reddit post titles discuss topics such as artificial intelligence, research, or emerging technologies that overlap conceptually between these two categories. For example, titles related to AI research or data analysis may be ambiguous without additional context.

The **books** category shows slightly lower recall compared to other classes. This can be attributed to the general and descriptive nature of book-related titles, which may lack distinctive keywords and resemble content from other categories. In contrast, categories such as **fitness** and **sports** achieve high precision and recall due to their more specialized and consistent vocabulary.

Overall, the observed misclassifications are semantically reasonable and reflect inherent ambiguity in short-text data rather than systematic model errors.

4.4 Statistical Stability and Significance Analysis

To assess the stability of model performance, **5-fold cross-validation** was applied to the Linear SVM model. Cross-validation results showed consistent accuracy across folds, with minimal variation. This indicates that the model's performance is not dependent on a particular train-test split and that the results are reliable.

Although formal statistical significance tests were not conducted for all model comparisons, the consistency between cross-validation accuracy and test accuracy suggests that performance differences are stable rather than random. The relatively small performance gap between models further indicates that improvements are incremental and depend largely on model choice and feature representation.

4.5 Discussion of Results

The experimental results demonstrate that both traditional machine learning and deep learning approaches are effective for short-text classification. Linear models perform strongly due to their compatibility with TF-IDF features, while the MLP provides a slight improvement by capturing non-linear relationships.

These findings highlight the importance of systematic evaluation and careful feature engineering. The results also show that simpler models can achieve competitive performance when properly tuned and evaluated, making them suitable for real-world text classification tasks.

5. Discussion

This section interprets the experimental results, analyzes model behavior, discusses limitations and future improvements, and summarizes key lessons learned from the project.

5.1 Interpretation of Results

The results show that all implemented models perform effectively on the task of multi-class Reddit post title classification. Traditional machine learning models, particularly Logistic Regression and Linear SVM, achieved strong performance when combined with TF-IDF features. This confirms that linear classifiers are well-suited for short-text classification problems involving high-dimensional sparse data.

The Multi-Layer Perceptron (MLP) achieved the highest accuracy, indicating that introducing non-linear modeling can provide a modest improvement over linear approaches. However, the performance difference between the MLP and Linear SVM is relatively small, suggesting that feature representation and data quality have a greater impact on performance than model complexity alone. Multinomial Naive Bayes also performed competitively, demonstrating that simple probabilistic models remain effective for text-based tasks.

5.2 Bias–Variance Analysis

Bias–variance behavior was examined using learning curves and cross-validation results. The learning curves showed that training accuracy remained high while validation accuracy gradually increased and converged toward 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 and do not suffer from significant overfitting. Regularization techniques and feature dimensionality control contributed to this balance. Cross-validation results further supported model stability, indicating that performance was consistent across different data splits.

5.3 Limitations and Future Work

Despite the strong results, the project has several limitations. First, the dataset consists only of Reddit post titles, which provide limited context and can lead to ambiguity between similar categories such as science and technology. Incorporating full post content could improve classification accuracy.

Second, the feature representations used 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 improve semantic understanding.

Additionally, more advanced deep learning architectures, such as convolutional or transformer-based models, could be explored. Expanding the dataset and evaluating the approach on other platforms could further improve model robustness.

5.4 Lessons Learned

This project demonstrated the importance of strong preprocessing and feature engineering in text classification tasks. One key lesson is that simpler models can achieve competitive performance when paired with effective feature representations. Another important insight is the value of systematic evaluation using cross-validation, learning curves, and error analysis to understand model behavior.

The project also highlighted the importance of reproducibility and collaboration in machine learning workflows. Maintaining a clear experimental pipeline and using version control tools helped ensure consistency and efficient teamwork.