News Classifier

Reza Omidvar

June 11, 2023

Contents

1 Executive Summary

The objective of this project was to develop a news classifier capable of accurately categorising articles into one of seven classes, with a particular focus on **world**, **politics**, and **science**. To achieve this, we followed a comprehensive approach that included data ingestion and enrichment, exploratory data analysis, model training and evaluation, and results visualisation.

We began by loading and preprocessing the provided dataset, which contained news articles and their corresponding categories. The data was enriched through text processing techniques, such as tokenisation, lemmatisation, and feature extraction using the TF-IDF method.

An exploratory data analysis was conducted to gain insights into the dataset and identify patterns or trends. This analysis included visualisations of class distribution and word clouds for each category.

For the model training and evaluation, we employed Logistic Regression, Random Forrest, Support Vector Machine and LSTM models. The models' performance were assessed using evaluation metrics such as F1 Score, accuracy, precision, and recall. The models also utilize techniques such as cross-validation, under-sampling, oversampling, and class weight optimisation.

The results' visualisation showcased the classifier's performance across all categories. Our model demonstrated strong performance in classifying news articles, and meeting the client's expectations.

In conclusion, the news classifier developed in this project has the potential to significantly benefit the client's business by automating the categorisation of news articles. The next steps include deploying the classifier, refining the model with additional data or algorithms, and addressing any limitations encountered during the project.

2 Introduction

In today's fast-paced digital world, staying informed and up-to-date with the latest news is crucial for businesses and individuals alike. Our client, recognising the importance of efficient news categorisation, has requested the development of a news classifier capable of accurately categorising articles into one of seven classes: world, politics, science, automobile, entertainment, sports and technology. The primary focus of this project is on the world, politics, and science articles, as these are of the greatest interest to the client. However, the classifier is expected to perform well across all categories.

The dataset provided by the client, data.csv, contains news articles and their corresponding categories. Our task is to build a machine learning solution that encompasses the following steps:

- **Data ingestion and enrichment:** Load and preprocess the dataset, applying any necessary text processing techniques to enhance the data quality.
- **Data visualisation:** Present visualisations of the dataset to provide insights into the data and its distribution across the seven classes.
- **Model training and evaluation:** Train a machine learning model to classify news articles and evaluate its performance using appropriate metrics.
- **Results visualisation:** Showcase the model's performance through visualisations and highlighting its effectiveness in classifying the articles.

This report will present our findings and insights from the project, as well as recommendations for next steps.

3 Data Ingestion and Enrichment

The first step in our project was to load and preprocess the dataset provided by the client. The dataset, **data.csv**, contained a total of 2318 news articles. Each article was characterized by three features: headline, article, and category. The headline and article

features contained the title and content of the news article, respectively, while the category feature represented the class label for each article.

During the data ingestion process, we performed an initial analysis to identify any missing values in the dataset. We found that there were 349 missing values across all features. To ensure the quality of our data, we decided to drop the rows containing these missing values.

Next, we focused on enriching the dataset through various text processing techniques. The following steps were taken to preprocess the text data:

Remove special characters and convert text to lowercase: We removed any special characters, such as punctuation marks, from the text and converted all characters to lowercase. This step helped to standardise the text and reduce noise in the data.

Tokenise the text: The text was tokenized into individual words, which allowed us to perform further text processing and feature extraction more effectively.

Remove stopwords: We removed common stopwords, such as "the", "and", and "in", from the text. Stopwords typically do not carry significant meaning and can be safely removed to reduce the dimensionality of the data.

Lemmatise the text: We applied lemmatisation to the text, which involved converting words to their base or dictionary form (e.g., "running" to "run"). This step helped to further reduce the dimensionality of the data and improve the performance of our machine learning models.

Remove numbers: We removed any numbers from the text, as they were not considered relevant to the classification task.

After completing the data ingestion and enrichment process, we were left with a clean and high-quality dataset, ready for further analysis and model training.

4 Exploratory Data Analysis

To gain a deeper understanding of the dataset and identify any patterns or trends, we conducted an exploratory data analysis. This analysis included visualisations of the distribution of categories, headline and article lengths, and word clouds for each category.

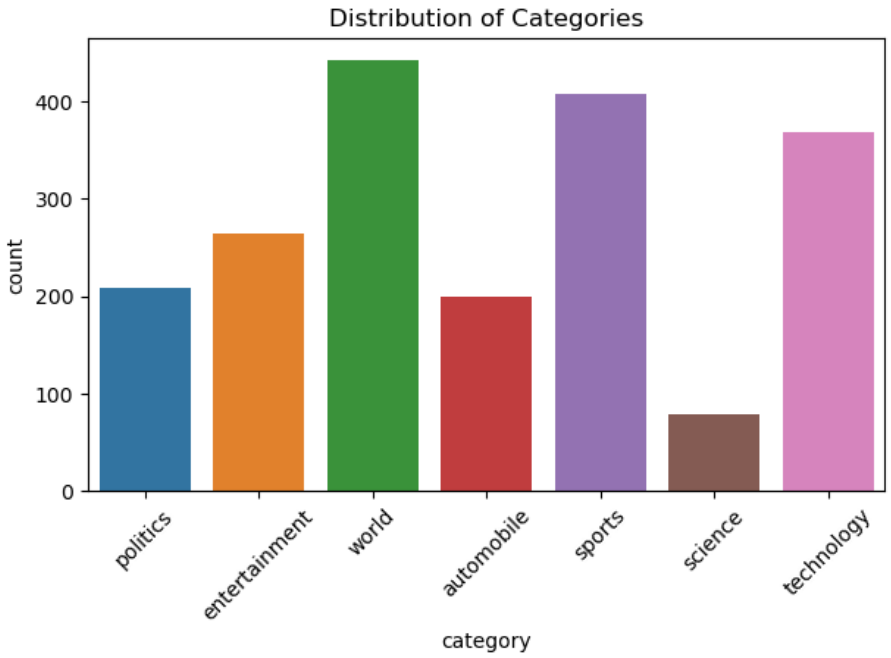


Figure 1: Distribution of categories in the dataset

4.1 Distribution of Categories

We began by visualising the distribution of categories in the dataset (Figure 1). The analysis revealed that the data was imbalanced, with a majority of articles belonging to the ‘world‘ and ‘sport‘ categories, while the ‘science‘ category was underrepresented. This imbalance could potentially impact the performance of our classifier, particularly for the minority class.

4.2 Headline and Article Lengths

We examined the distribution of headline and article lengths across the dataset (Figure 2). This analysis provided insights into the structure of the news articles and helped us understand the variability in the text data.

4.3 Word Clouds

We generated word clouds for each category to visualise the most frequently occurring words (Figure 3 a-g). For example, we observed that ”Film” was dominant in the ‘entertainment‘ category, ”China” in the ‘world‘ category, and ”Tesla” in the ‘automobile‘

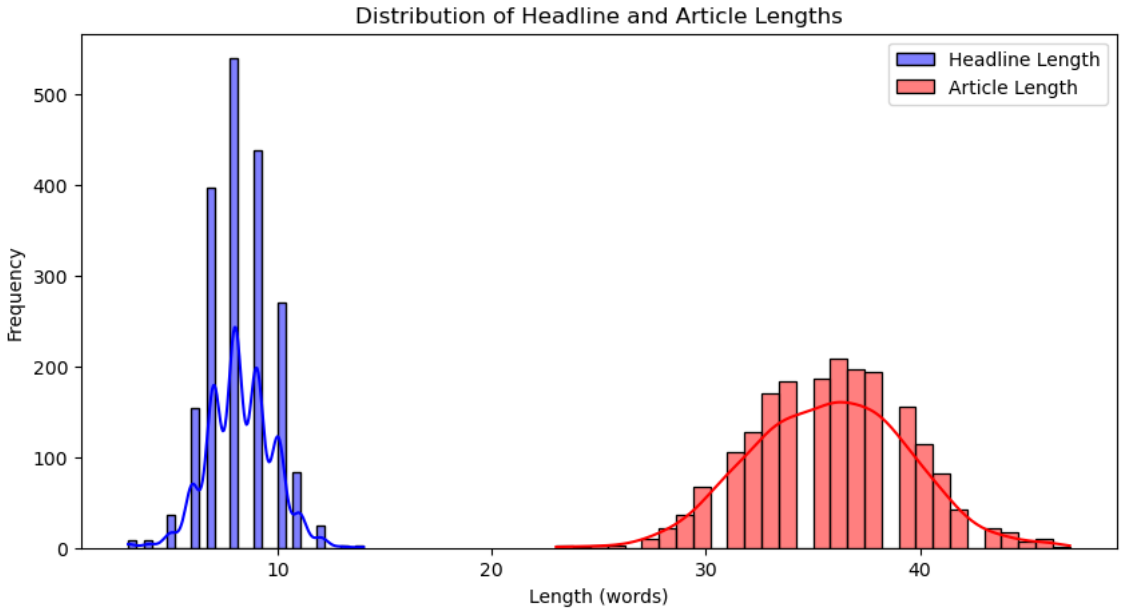


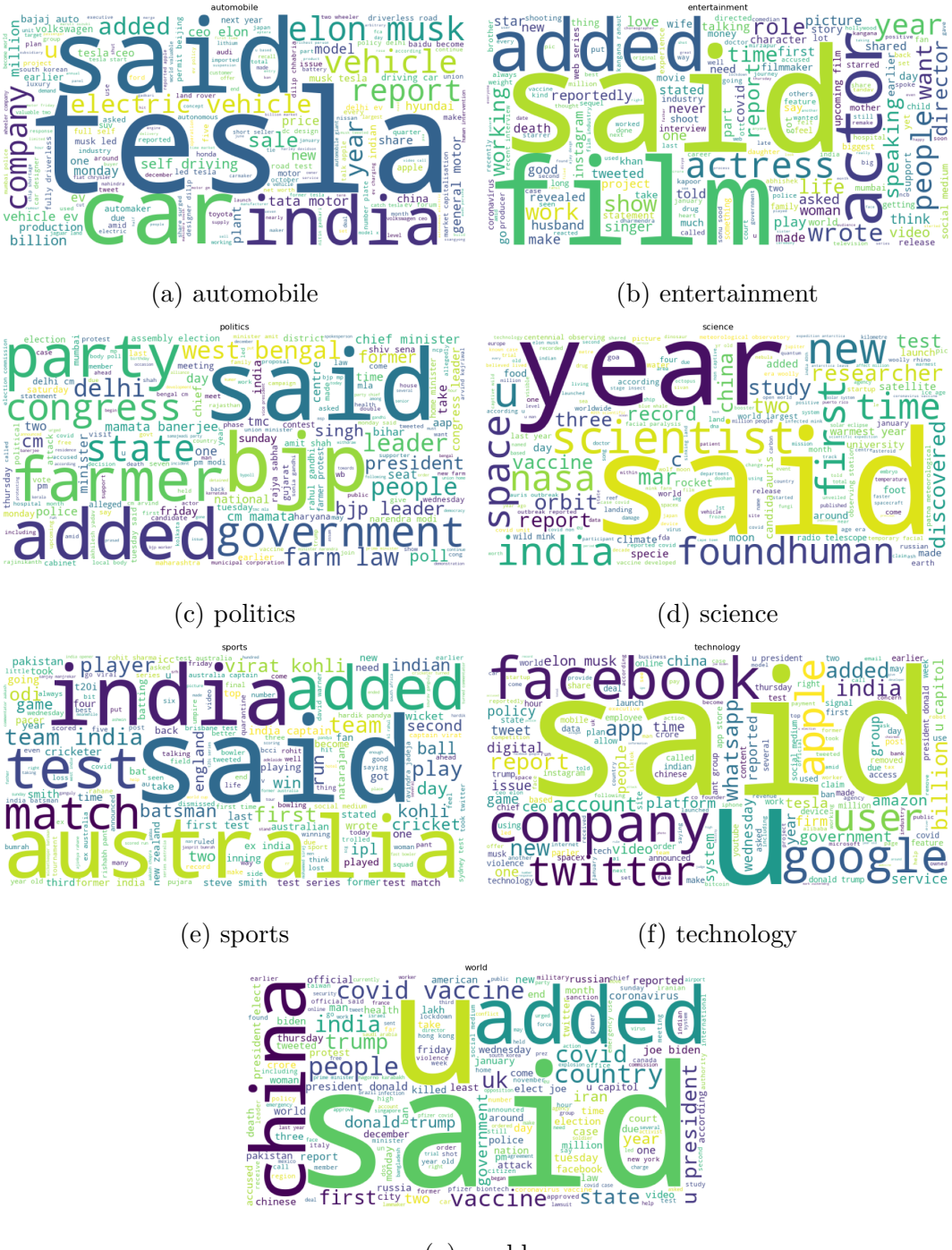
Figure 2: Distribution of headline and article lengths

category. Interestingly, we also found that some words, such as "said", were repetitive across most categories and did not provide additional information for classification. To address this issue, we had two options: add these words to the list of stopwords and remove them from the text, or reduce their impact by using TF-IDF tokenisation. We chose the latter approach, as it allowed us to retain potentially useful information while minimizing the influence of less informative words.

4.4 Cleaned Data

Based on the intuition that headlines are often more descriptive of the news category and to provide more text data for the model, we combined the 'headline' and 'article' columns into a single column. This step ensured that both the headline and article content were considered during the classification process.

After completing the exploratory data analysis, we saved the cleaned and processed dataset for use in the model training phase.



(a) automobile

(b) entertainment

(c) politics

(d) science

(e) sports

(f) technology

(g) world

Figure 3: Word clouds for each category

5 Model Training and Evaluation

With the clean and processed dataset ready, we proceeded to the model training and evaluation phase. We specified a configuration file for certain setups, such as the n-gram range, to ensure consistency throughout the process.

We trained our model on different types of datasets to compare their performance. The datasets included:

1. The whole dataset, which served as the benchmark for training.

2. An undersampled dataset, where the number of samples in each category was equalized by reducing the majority classes.

3. An oversampled dataset, generated using the Synthetic Minority Over-sampling Technique (SMOTE)[1]. SMOTE is an oversampling technique that creates synthetic samples for the minority class to balance the dataset.

In addition, to maintain the ratio of classes in each fold during the model training process, we employed stratified k-fold cross-validation. This technique ensured that each fold had a representative distribution of the classes.

We used the TF-IDF vectorization method to convert the text data into numerical features, making it suitable for machine learning algorithms. We trained our model using various algorithms, including Logistic Regression, Random Forest, and Support Vector Machine (SVM). Additionally, we trained a Long Short-Term Memory (LSTM) model, which is a type of recurrent neural network that can selectively remember or forget learnings.

Once the models were trained, we evaluated their performance using various metrics, such as accuracy, confusion matrix, precision, recall, and F1 score. These metrics provided a comprehensive assessment of each model's ability to correctly classify news articles across all categories, with a particular focus on the 'world', 'politics', and 'science' classes.

To overcome the problem of imbalanced datasets, we utilized class weight optimization. This technique allowed the models to pay more attention to the minority classes, where their performance was not as good.

By comparing the performance of the different models and dataset configurations, we were able to identify the most effective approach for our news classifier. This information will be invaluable in guiding future improvements and refinements to the model.

Table 1: Accuracy and F1 score for models trained on the benchmark dataset

Model	**Accuracy**	**Average F1 Score**
Logistic Regression	0.82	0.76
Random Forest	**0.82**	**0.78**
SVM	0.80	0.77
LSTM	0.62	0.47

6 Results Visualisation

To effectively communicate the performance of our news classifier, we created visualisations that showcased the model's performance across all categories. We present the results for the benchmark, oversampled, and class weight optimized datasets.

6.1 Benchmark Results

Table 1 shows the accuracy and F1 score for the models trained on the benchmark dataset. As can be seen, Random Forest performed the best with an accuracy of 82% and an average F1 score of 0.78. The confusion matrix for Random Forest is shown in Figure 4. It is interesting to note that the model works well in the 'world' and 'politics' categories, but not as well in the 'science' category. Additionally, there are many observations misclassified as 'world', indicating that the content of news in this category is quite general and similar to other classes.

6.2 Oversampled Results

Moving on to the oversampled dataset, we observed a significant improvement in the performance of the Logistic Regression model across all categories, as shown in figure 5. Specifically, the F1 score for the 'science' category improved from 0.38 to 0.45.

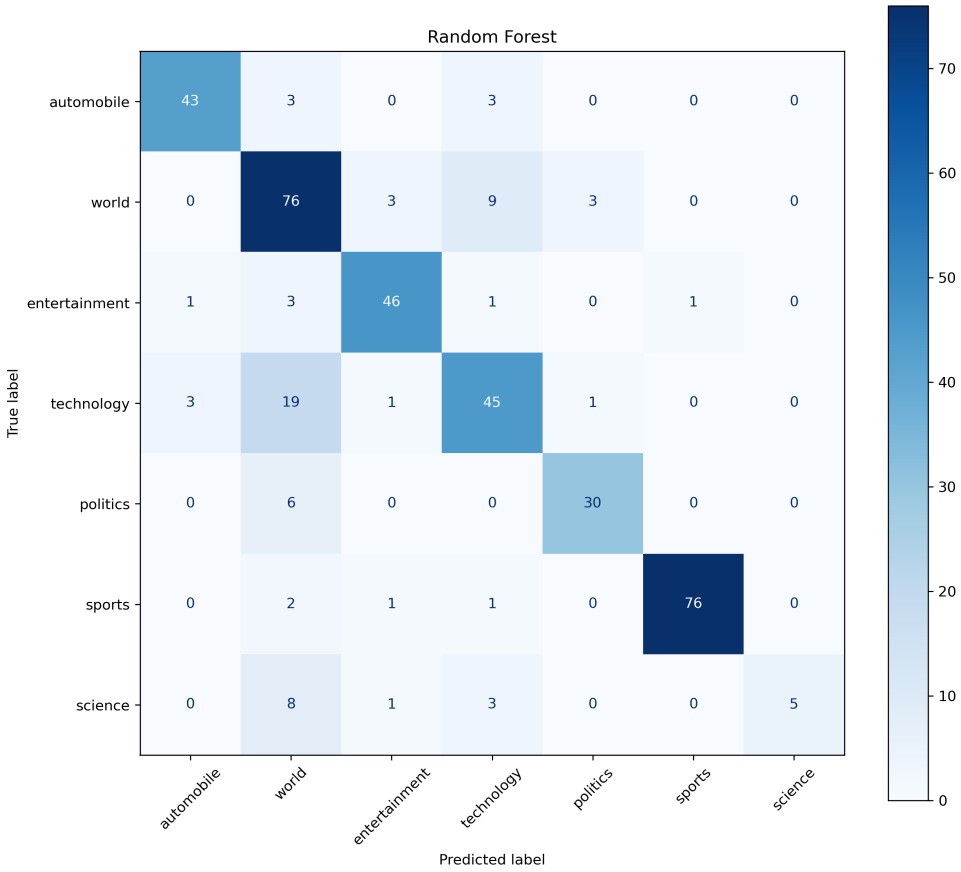


Figure 4: Confusion matrix for the Random Forest model on the benchmark dataset

6.3 Class Weight Optimized Results

Using class weight optimization to penalize the model based on the classes resulted in an improvement better than oversampling specifically in the Logistic Regression model as shown in figure 6.

6.4 LSTM

We also generated plots for loss and accuracy (Figures 7 a and b) to visualize the performance of the LSTM model. Early stopping was used to avoid overfitting problems. The LSTM model also showed improvement when using class weight optimization.

	index	precision	recall	f1-score	support	Evaluated_model
0	automobile	0.907407	1.000000	0.951456	49.000000	Logistic Regression
1	world	0.778947	0.813187	0.795699	91.000000	Logistic Regression
2	entertainment	0.905660	0.923077	0.914286	52.000000	Logistic Regression
3	technology	0.776119	0.753623	0.764706	69.000000	Logistic Regression
4	politics	0.891892	0.916667	0.904110	36.000000	Logistic Regression
5	sports	0.987179	0.962500	0.974684	80.000000	Logistic Regression
6	science	0.800000	0.470588	0.592593	17.000000	Logistic Regression
7	accuracy	0.865482	0.865482	0.865482	0.865482	Logistic Regression
8	macro avg	0.863887	0.834235	0.842505	394.000000	Logistic Regression
9	weighted avg	0.864660	0.865482	0.862777	394.000000	Logistic Regression

Figure 5: Classification Report for the Logistic Regression model on the SMOTE Oversampled dataset

6.4.1 Overall Results

Our results demonstrate that the Random Forest and Logistic Regression models performed well in classifying news articles across all categories. Additionally, using oversampling and class weight optimization techniques improved the performance of the models, particularly for the minority classes.

7 Recommendations and Next Steps

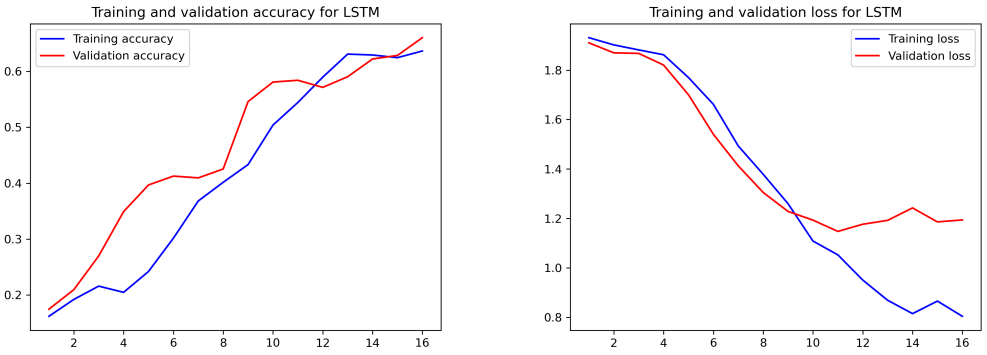
Based on our analysis and the performance of the various models, we provide the following recommendations and next steps to further improve the news classifier:

1. **Fine-tuning Hyperparameters:** Continue fine-tuning the hyperparameters of the models to optimize their performance. Techniques such as grid search and random search can be employed to systematically explore the hyperparameter space.

2. **Deep Learning Models:** Explore more advanced deep learning models, such as BERT or Transformer-based architectures, which have shown promising results in text classification tasks. These models can potentially capture more complex patterns in text data and improve classification performance.

	index	precision	recall	f1-score	support	Evaluated_model
0	automobile	0.907407	1.000000	0.951456	49.000000	Logistic Regression
1	world	0.771739	0.780220	0.775956	91.000000	Logistic Regression
2	entertainment	0.842105	0.923077	0.880734	52.000000	Logistic Regression
3	technology	0.793651	0.724638	0.757576	69.000000	Logistic Regression
4	politics	0.868421	0.916667	0.891892	36.000000	Logistic Regression
5	sports	0.987179	0.962500	0.974684	80.000000	Logistic Regression
6	science	0.750000	0.529412	0.620690	17.000000	Logistic Regression
7	accuracy	0.855330	0.855330	0.855330	0.855330	Logistic Regression
8	macro avg	0.845786	0.833788	0.836141	394.000000	Logistic Regression
9	weighted avg	0.853376	0.855330	0.852637	394.000000	Logistic Regression

Figure 6: Classification Report for the Logistic Regression model with class weight optimisation



(a) Accuracy

(b) Loss

Figure 7: Accuracy and Loss curves for the LSTM model

3. **Model Interpretability:** Investigate model interpretability techniques, such as LIME or SHAP, to gain insights into the decision-making process of the models. This can help identify areas for improvement and provide a better understanding of the factors influencing the classification results.

4. **Multilabel Classification:** Consider using a multilabel dataset and adapting the classifier to handle multilabel classification problems. This approach can account for articles that belong to multiple categories simultaneously, providing a more accurate representation of the complex relationships between news topics.

5. **Regular Model Updates:** Periodically update the models with new data to ensure that they remain relevant and accurate as the news landscape evolves. This can help maintain the classifier's performance over time.

6. **User Feedback:** Implement a user feedback mechanism to collect information on the classifier's performance in real-world scenarios. This feedback can be used to identify areas for improvement and guide future refinements to the model.

By implementing these recommendations and continuously refining the news classifier, we can ensure that it remains an effective and valuable tool for categorizing news articles across various categories.

8 Conclusion

In this project, we developed a news classifier capable of categorizing news articles across various categories, such as 'world', 'politics', and 'science'. We conducted an exploratory data analysis to gain insights into the dataset and preprocessed the text data to prepare it for model training. We experimented with different machine learning algorithms, including Logistic Regression, Random Forest, Support Vector Machine, and LSTM, and employed techniques such as oversampling, undersampling, and class weight optimization to address the imbalanced dataset issue.

Our results demonstrated that the models performed well in classifying news articles, with the Random Forest and Logistic Regression models showing particularly strong performance. We also observed improvements in the models' performance when using oversampling and class weight optimization techniques.

This classifier can be utilized by news organizations, content curators, and researchers to better understand and organize the vast amount of news content generated daily, ultimately providing a more streamlined and efficient way to access and analyse news data.

References

[1] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011.