The Article Analyst

Machine Learning Insights Into New York Times Journalism

Qian Fu
Kevin Deloria
Yash Dave

Introduction

In the digital age, the exponential increase in text data, particularly from news sources, presents both challenges and opportunities for automated processing and analysis. The New York Times (NYT), as a prolific source of news articles, offers a rich dataset for exploring advanced machine learning techniques. Motivated by the potential to improve the accessibility and understanding of news content, our research embarked on a journey to uncover the thematic structures inherent in the news articles and to predict their topics with high accuracy. The problem at the heart of our research revolves around the efficient and accurate categorization of news articles into coherent topics, a task that is fundamental for information retrieval, content summarization, and recommendation systems. Solving this problem not only aids in better information navigation for users but also provides insights into the prevailing narratives within the media landscape. In the unsupervised phase, we applied Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF) to distill topics from the articles. Our supervised learning phase built on this foundation, utilizing Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Random Forests (RF) to classify articles into the topics identified earlier.

From the topic modeling task, we observed the persistent emergence of political topics, with significant features including terms such as "state," "government," and "president." When classifying the topics, we found that all models achieved remarkable accuracy, despite the dataset's imbalance. Crucially, besides the main predictive feature, "text," features like "news desk" and "section name" were identified as key predictors in topic classification. This underscores the importance of metadata in thematic analysis. Our study uniquely demonstrates the effectiveness of NMF over LDA in generating coherent news topics and introduces a pioneering approach by incorporating metadata features like "news desk" and "section name" into topic classification models. These advancements not only enhance classification accuracy but also open new pathways for combining textual and non-textual data in machine learning, promising to enrich future text analysis methodologies.

Related Works

When we were exploring the topic of using machine learning methods with New York Times articles, we did extensive research on similar approaches that had already been done in order to inform and inspire our project direction. The first related work is a Medium article by Hedi Manai titled “Machine Learning: Text Classification of News Articles”[6] . This article describes the steps taken by the author in a project that used text classification to identify the category of newspaper articles. It details the steps from data cleaning and preprocessing to feature engineering and supervised learning model training and selection. This work is quite similar to our project, as we are taking on the same task of classifying articles via text classification. However, our approach varies in that we used unsupervised learning techniques to perform topic modeling in order to categorize our articles, and then based on those identified topics we performed supervised learning. A second related work is an article on Medium titled “Unsupervised Text Classification with Topic Models and Good Old Human Reasoning” by Márton Kardos[5] . This article goes over a project which uses the sklearn 20newsgroups data set to perform topic modeling using NMF, and then performing classification using topicwizard. While our project will not be using this specific approach to topic modeling, this project gave us useful insight into the use of NMF in our unsupervised learning approach. The third related work we identified was a Medium article titled “BBC News Text Classification” by Cigdem Tuncer[12] . In this article the author discusses the data preprocessing and supervised learning steps taken to classify BBC News articles. This article provides useful information about the supervised learning steps we can take. However, our project differs in that we utilized both supervised and unsupervised learning approaches in order to model topics and then classify the New York Times articles.

Data Source

The dataset we are using for this project, applicable to both supervised and unsupervised learning tasks, was retrieved from the NYT API, and we have stored it in our GitHub repository. It encompasses articles from four news agencies: New York Times, AP, Reuters, and International Herald Tribune, exclusively focusing on articles and excluding other document types. The format of the data returned from the API was a dictionary, which we subsequently stored in a CSV format.

Initially, we retrieved 8,621 articles featuring 21 attributes. Following a cleaning process, where we eliminated some unnecessary columns and those with a majority of missing values, and also dropped all rows with missing values, the dataset was reduced to 7,064 instances with 15 attributes. After exploring the cleaned dataset, we decided to retain 8 features for use in both supervised and unsupervised learning tasks. Out of these 8 features, the most important variables we identified after exploration are: "abstract," "lead_paragraph," "headline," and "keywords." The content of these four features can closely reflect the overarching topic of the article. For detailed information on each feature of the dataset, please refer to **Appendix 1.1.**

Feature Engineering

- Unsupervised Learning:

  For the eight features retained for use in the project's tasks. Four of these features contain content that closely reflects the overall topic of the article. We combined these four features into a single feature named "combined_text," which was the only feature used in our unsupervised learning task. To prepare the "combined_text," feature for use in our topic modeling models, we have executed the following additional preprocessing steps: removal of special characters, punctuation, and numbers; conversion of text to lowercase; tokenization of the text into individual words; elimination of stopwords; and application of lemmatization to reduce words to their root forms. For different topic modeling methods, we selected appropriate text vectorization techniques: Bag of Words (BoW) for Latent Dirichlet Allocation (LDA) and Term Frequency-Inverse Document Frequency (TF-IDF) for Non-negative Matrix Factorization (NMF).

- Supervised Learning:

  We used a total of six features in our topic classification task. "Topic_label" is the target variable, which represents the topic generated from our topic modeling task for each article. "Combined_text" serves as the primary predictor; it is a textual feature processed using the same preprocessing steps as in the unsupervised learning part. The other four features are non-textual and include categorical data selected as helper predictors. The preprocessing steps executed to prepare these four features for our classifiers involved extracting key values from features where the value is a list of dictionaries, consolidating rare categories and eliminating infrequent ones. To avoid causing high dimensionality from one-hot encoding the four categorical features, we selected only the top 10 categories for each feature based on their frequency, grouping the less frequent categories into an "other "category. After a series of preprocessing steps, we applied one-hot encoding to these four features. For the complete list of all final features, please refer to **Appendix 1.2.**

For which notebooks were used for each part, please refer to Appendix 1.3.

Part A. Unsupervised Learning

Methods Description and Evaluation Metrics Justification

In our topic modeling endeavor, we began with Latent Dirichlet Allocation (LDA) for its flexibility and robustness in handling various kinds of text data without requiring prior labeling, making it ideal for initial dataset exploration to uncover a broad spectrum of topics. We then applied NMF, known for its ability to generate more interpretable and distinct topics. This characteristic is particularly beneficial for our topic classification task, as it minimizes topic overlap. The decision to use both methods stemmed from their complementary strengths: LDA's capacity for broad topic identification and NMF's precision in topic distinction. Our analysis involves comparing the performance of both methods using the 'combined_text' feature(for feature representation and feature engineering, please refer to the "Feature Engineering" section) to determine the optimal approach for topic classification. We anticipate further refining our preprocessing steps to enhance the selected model's effectiveness.

For both methods, we use both qualitative and quantitative aspects of model evaluation. For each model configuration, we examine the top 50 words in each topic to assess the distinctiveness of the topics by checking for common words that might appear across multiple topics, which could make the topics less distinct. We utilize the coherence score as a quantitative metric to evaluate the quality of the topics generated by the model. A higher coherence score typically indicates that the topics are more meaningful and interpretable. Additionally, we visualize the similarity between topics using a heatmap, which aids in identifying how topics relate to each other and whether there are any overlaps that might affect the distinctiveness of the topics. Lastly, we visualize how articles are distributed across the topics. This analysis helps us understand the coverage of topics in our dataset and see if there are any dominant or sparse topics. Assessing the balance and representativeness of topics is crucial to ensure a well-performing classification model.

Method One: Latent Dirichlet Allocation (LDA)

We focused our exploration on the effects of varying the number of topics in the model, assigning 50 iterations for convergence while maintaining default settings for all other parameters. We began with 10 topics to gain a detailed understanding of our dataset's structure. This initial phase aimed at uncovering a wide range of themes, capturing the dataset's nuanced details. We then reduced the number of topics to 5 to observe changes in topic coherence and distinctiveness. This step also allowed us to assess the distribution balance of articles across topics, essential for developing a robust classification model. Finally, we condensed the topics to 2, aiming to understand the broadest categorizations within our articles.

Analysis of Evaluation Results and Derived Insights:

For each model—the one with 10 topics, the one with 5 topics, and the one with 2 topics—we extracted the top 50 words from every topic and used these terms to manually assign a descriptive label to each. The table below lists the topic labels for each model:

Table 1:

	Descriptive Labels
10-Topic Model	Topic 1: NYC Local News and Culture Topic 2: Real Estate and Environmental Concerns Topic 3: US Politics and International Affairs Topic 4: Arts, Culture, and Obituaries Topic 5: Literature and International News Topic 6: Economy, Healthcare, and Labor Topic 7: College Sports and Education Topic 8: Entertainment and Media Topic 9: Crime, Law Enforcement, and Justice Topic 10: Social Announcements and Personal Stories
5-Topic Model	Topic 1: Culture and Entertainment Topic 2: Urban Life and Business Topic 3: Politics and Government Topic 4: Crime and Justice Topic 5: Sports and Personal Events
2-Topic Model	Topic 1: Politics, Governance, and Social Issues Topic 2: Culture, Lifestyle, and Education

Here are our findings after analyzing the top 50 words of each topic and comparing the descriptive label of each topic generated by the three models.:

The common words across various topics and models remain largely the same: "new," "year," "city," "state," and "people". "The consistency of these common words also suggests that there is an opportunity to further refine our preprocessing steps by removing additional stopwords.

As the number of topics decreases, the topics become broader and more general, encompassing a wider range of subjects within each topic. The 10-topic model provides more specific and focused themes, while the 2-topic model results in very broad categories that aggregate diverse content.

Despite the reduction in topics, certain themes such as politics and entertainment persist across models, which speaks to the salient nature of these subjects within the dataset.

Coherence Scores:

Table 2:

	10-Topic Model	5-Topic Model	2-Topic Model
Coherence Score	0.495	0.465	0.417

Here are the key insights derived from the coherence scores of each model:

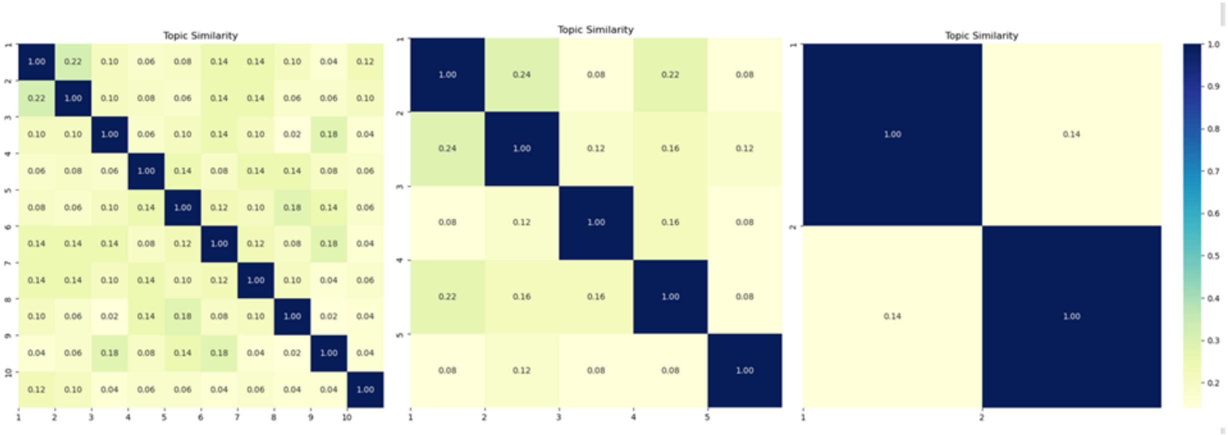
The 10-topic model has the highest score (0.495), indicating that the topics are the most meaningful and interpretable.

Reducing topics to 5 slightly lowers coherence (0.465), hinting at a minor loss in topic clarity by merging related themes.

The largest drop occurs with the 2-topic model (0.417), indicating that broader categorizations significantly diminish meaningfulness and interpretability, as they amalgamate a wide range of content into too few topics.

Topic Similarity:

Figure 1



Interpretation and Key Findings:

For the 10-topic model, we see mostly light-colored cells, indicating that most topics have low similarity scores with each other, which suggests good topic separation.

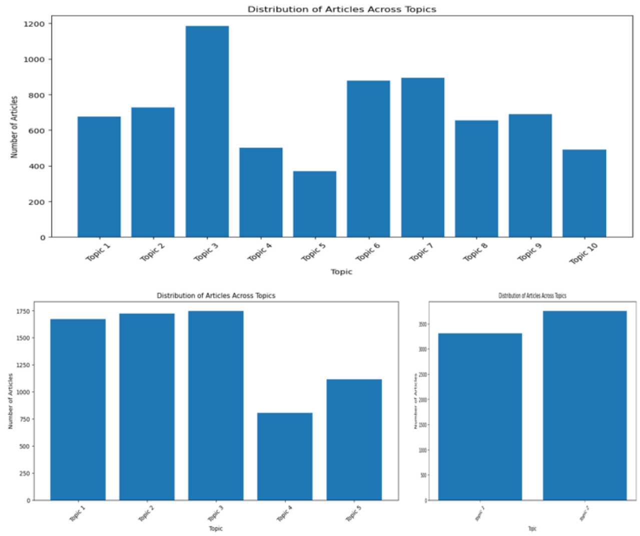
For the 5-topic model shows that the topics are generally distinct, However, there's a noticeable similarity between topics 1 and 2, as indicated by the darker cell. This suggests that in reducing the number of topics from 10 to 5, some topics have been merged, which could lead to a certain degree of overlap in the content they cover.

The 2-topic model heatmap presents a very stark contrast, with only two topics to compare. This is expected because with only two topics, each topic is likely to be very broad and encompass a wide range of diverse content.

Article Distribution Across Topics:

Our key findings highlight uneven article distribution across models:

Figure 2



The 10-topic model shows significant imbalance, with some topics, like Topic 3, dominating.

The 5-topic model reduces this disparity, indicating a more balanced distribution.

The 2-topic model achieves an almost even distribution, dividing content into two broad categories. However, this near-equal split in the 2-topic model, while useful for high-level tasks, may miss finer thematic nuances.

Method Two: Non-negative Matrix Factorization (NMF)

Through the exploration with the LDA model, we have gained a comprehensive understanding of the overall thematic structure within our dataset. hi, we now turn our attention to the NMF model.

We began directly with 2 topics, and surprisingly, the coherence score of the 2-topic NMF model exceeded 0.5, reaching approximately 0.548. When exploring LDA models, we initially decided to examine models with 10, 5, and 2 topics. For the NMF model, after starting with 2 topics, we conducted tuning of the number of topics to identify the configuration with the highest coherence score. This chart shows that the configuration with the highest coherence score features 3 topics, achieving a score of 0.65. It is quite exciting that, with only 3 topics, we achieved such a high coherence score. Also, upon tuning the number of topics from 2 to 10 for the NMF models, we observed that the coherence scores consistently surpassed those of the LDA models. This suggests that NMF, across a range of topic quantities, provides a higher level of topic coherence compared to its LDA counterparts.

Based on the findings from exploring the LDA models, we refined the preprocessing of the 'preprocessed_text' feature by removing some common words: "new," "york," "year," and "city." After vectorizing this newly preprocessed data, we applied the NMF model with 3 topics and observed an increase in the coherence score from 0.65 to 0.71. This indicates that refining the preprocessing step indeed enhances our model's performance.

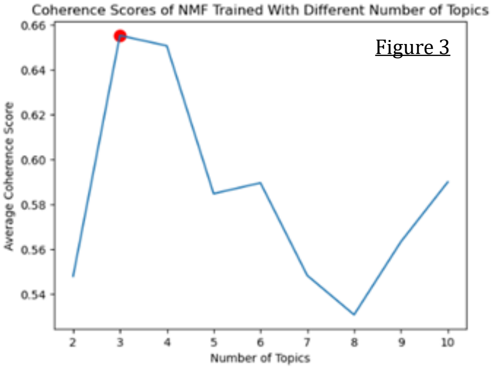
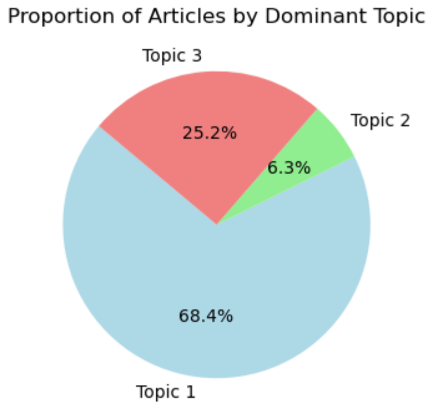


Figure 3

As now we can see that the NMF method seems to provide coherent and distinct topics that are well-suited for classification purposes, with the ability to maintain specificity even when the number of topics is low. The 3-topic model, in particular, achieves a balance by offering distinct thematic categories that could potentially enhance the performance of a classification model due to their clear differentiation.

Now, let's specifically examine the distribution of articles across the three topics generated by the NMF model. As we can see here, there is a noticeable imbalance in the distribution of articles among the three topics. Topic 1 has the highest number of articles, indicating that it is the most prevalent theme within the dataset. Topic 2 has the fewest articles, which could suggest that it represents a more niche or specific theme compared to the others.

Figure 4



The significant difference in the number of articles across topics could lead to potential classification bias. The distribution reflects a diversity within the dataset, with certain themes being more represented than others. This could also affect the granularity of the topics; broader topics tend to encompass more articles.

Best Model Justification

Drawing on the insights from the LDA models, we've gained an understanding of the underlying themes within our dataset. Our main criterion for evaluating models is the coherence score. The table below lists out all the models we have explored and their coherence scores for comparison.

Table 3:

LDA Models: exploring the overall thematic structure of the dataset.	**Coherence Scores**
10-Topic Model	**0.495**
5-Topic Model	**0.465**
2-Topic Model	**0.417**
NMF Models: generating more interpretable and distinct topics	
2-topic model	**0.548**
3-topic model trained on original preprocessed data	**0.655**
3-topic model trained on refined preprocessed data.	**0.713**

For the 'best' model, we aim for the highest coherence score while limiting the number of topics to three, as these topics will be used for subsequent classification tasks. Overall, the NMF model provides coherent and distinct topics that are well-suited for classification purposes. Therefore, we have selected the NMF model with 3 topics as our 'best' model. Despite the significant imbalance in article distribution across these topics, we plan to employ technical solutions to address this issue during the training of our classification models.

Sensitivity Analysis: Non-Matrix Factorization

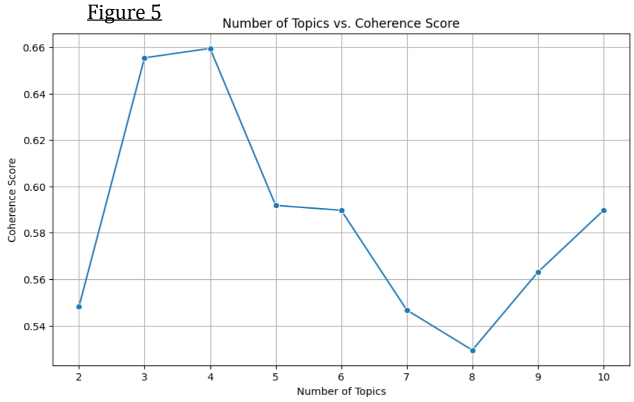
We approached sensitivity analysis on our best model by observing the effect of different hyperparameters on coherence scores.

First, we tune the model using the init, beta_loss, and solver hyperparameters. We have observed the coherence scores reveal a surprising consistency in coherence scores across different combinations despite systematically varying these parameters.

This suggests that the model exhibits insensitivity to changes in certain hyperparameters within the explored range.

Finally, we tune the model using n_components or

Figure 5



the number of topics keeping every other parameter consistent. As the number of topics increases, we observe a non-linear trend in the coherence score.

As we increase the number of topics from 2 to 4, there is a significant improvement in the coherence score. However, beyond 4 topics, further increasing the number of topics does not significantly improve coherence and may even result in a slight decrease in coherence.

Part B. Supervised Learning

Methods Description

In our topic classification task, we aimed to accurately categorize articles into one of three topics: Events, Lifestyle, and Politics, identified by our topic modeling. The distribution of articles showed a significant imbalance, with 68.4% of articles in Lifestyle, 25.2% in Politics and only 6.3% in Events. Additionally, feature engineering expanded our dataset to a high-dimensional space with 1065 columns (for feature representation and feature engineering, please refer to the "Feature Engineering" section.), complicating model selection and training.

Given the imbalance and high dimensionality of our dataset, we have conducted research on suitable models based on the nature and size of our data. we selected three methods:

Multinomial Naive Bayes (MNB): Chosen as our baseline due to its simplicity and efficiency in handling high-dimensional text data. Despite not directly addressing class imbalance, MNB's effectiveness in text classification provides a solid starting point for comparison.

Support Vector Machine (SVM): Selected for its capability to manage high-dimensional feature spaces and mitigate class imbalance effects.

Random Forest (RF): Opted for its robustness in handling class imbalance and reducing overfitting through its ensemble approach. RF's ability to perform feature importance analysis further aids in understanding and improving our model's performance.

After exploring the three methods and comparing their evaluation results, we will make our decision on the best model based on both performance metrics and practical considerations.

Model Exploration and Evaluation

In our model exploration, we began by fitting each selected model—MNB, SVM, and RF—using default parameters on the original, imbalanced dataset. To address the challenge of class imbalance, we applied two data balancing techniques: SMOTE and stratification. After balancing the data, we observed improved performance scores across all models, indicating the effectiveness of these techniques. Despite the initial success with default parameters, which already yielded excellent performance, we decided to explore further enhancements through hyperparameter tuning for some models. This decision was made to investigate whether adjustments could lead to even better results. Due to the extreme imbalance in our data, we have decided not to rely on the accuracy metric. Instead, we are focusing more on precision, recall, F1-score, and the ROC AUC score. We have chosen the ROC AUC score as our primary evaluation metric because it is less influenced by the class imbalance and provides a more nuanced understanding of model performance across different threshold settings. Consequently, we have conducted 5-fold cross-validation focusing on the ROC AUC score to ensure the robustness and reliability of our evaluation. The table below outlines the models we examined and their corresponding mean CV ROC AUC scores.

Table 4:

Models	**Mean CV ROC AUC Score**
Default MNB model trained on original dataset	0.959940262420897
Default RF model trained on original dataset	0.9902535988540133
Default SVM model trained on original dataset	0.9895296833713028
Default MNB model trained on stratified dataset	0.9593152904685829
Default RF model trained on SMOTE-processed dataset	0.9985219054015179
Default SVM model trained on SMOTE-processed dataset	0.9990913164415366
Default SVM model trained on stratified dataset	0.9906330933858966
Tuned MNB model trained on stratified dataset	0.9613090635926854
Tuned RF model trained on stratified dataset	0.998973128048061
Tuned RF model trained on SMOTE-processed dataset	0.9907826825983337

As we can observe from the table, all default models trained on the original dataset demonstrate strong performance, with the RF model achieving the highest score of 0.9903, and the MNB model having the lowest score of 0.9599. The lowest score is notably high, and the differences between the models are very minor. This indicates that, even without data balancing techniques, the models are quite capable of handling the imbalanced dataset to some extent, which is a significant surprise.

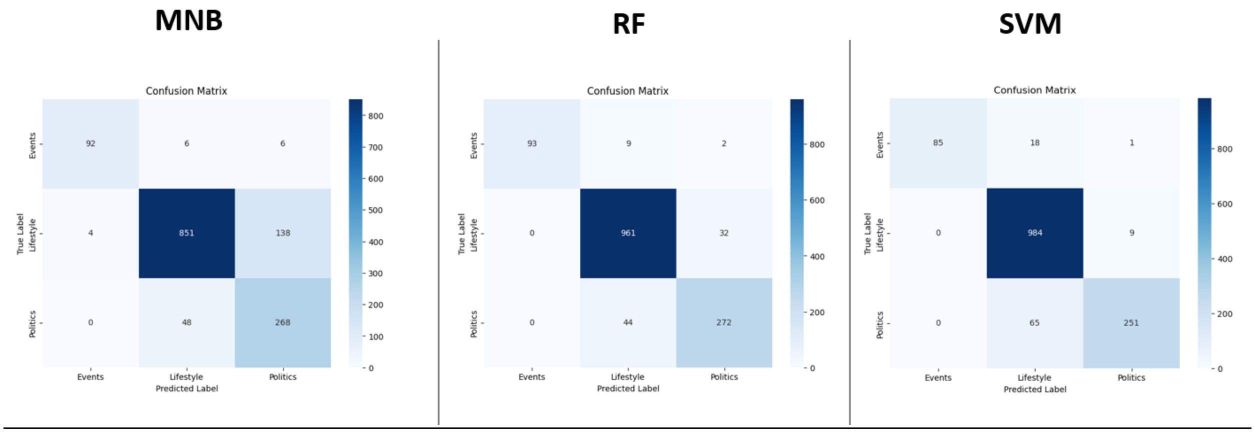
The data balancing techniques we applied notably improved the performance of the RF and SVM models, with scores increasing to nearly perfect at 0.9991. This suggests that SMOTE is particularly effective in enhancing model performance for these algorithms in the face of class imbalance.

When comparing the stratification data balancing method on the MNB and SVM models, it slightly decreased the performance of the MNB model but improved the SVM model's performance.

Upon exploring model tuning, we chose to tune the MNB model trained on the stratified dataset, which showed a slight improvement. The tuned RF model exhibited different behaviors based on the data preprocessing technique: it improved slightly on the stratified dataset to 0.9990 but decreased on the SMOTE-processed dataset to 0.9908. This suggests that the benefits of hyperparameter tuning may vary depending on the preprocessing technique and the intrinsic characteristics of the dataset. Given that the default SVM model already reached a near-perfect score of 0.999, we did not tune any SVM models.

For other performance metrics, such as precision, recall, and F1-score for all three classes of each model, please refer to **Appendix 2.1**. Next, we compare the ROC AUC curve and the confusion matrix of the default MNB model, default SVM model, and default RF model trained on the original data. For the ROC AUC curves and confusion matrices of all other models, please refer to the corresponding notebooks listed in **Appendix 1.3**.

Figure 6:



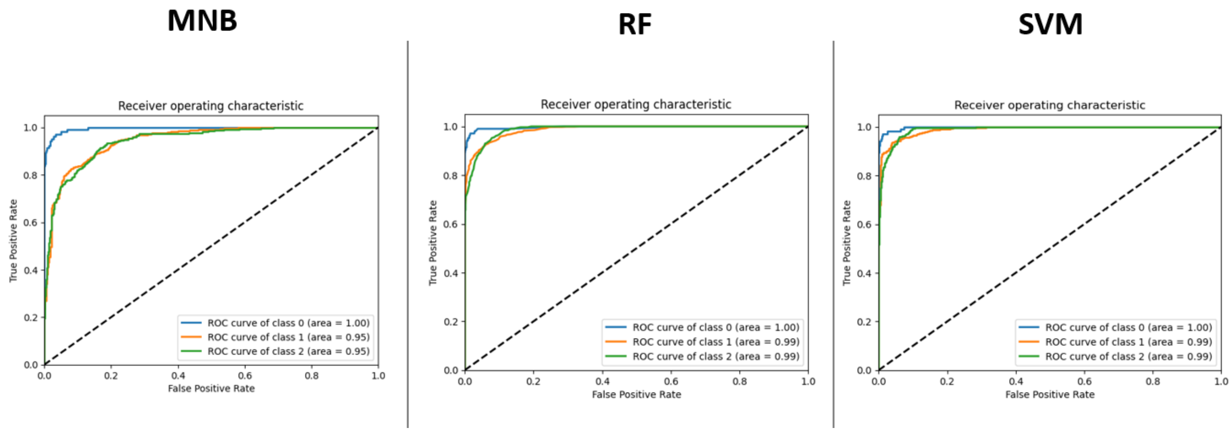
The above three confusion matrices for the three models provide insight into each model's performance across the three topics: Events, Lifestyle, and Politics. Here are the findings:

Table 5:

MNB	RF	SVM
· Shows a relatively balanced performance across the three categories. · It correctly classifies many 'Lifestyle' articles but also misclassifies 'Politics' articles as 'Lifestyle' to some extent. · 'Events' and 'Politics' have lower true positives compared to 'Lifestyle', but 'Events' are rarely misclassified as 'Politics' or vice versa.	· Exhibits the highest number of correct classifications for 'Lifestyle' and a significant improvement in correctly classifying 'Politics' articles compared to MNB. · There is a minor misclassification between 'Events' and 'Politics'. · It has a perfect classification for 'Lifestyle', indicating that none of the 'Lifestyle' articles were misclassified as 'Events' or 'Politics'.	· Demonstrates the best performance among the three models, with the highest number of correct predictions for 'Lifestyle' and 'Politics' and minimal misclassifications across all categories. · 'Events' has a slightly higher misclassification rate into 'Lifestyle' than the RF model but less than MNB. · It almost perfectly distinguishes between 'Lifestyle' and 'Politics', which suggests a strong feature separation capability.

In summary, while all models perform well for the 'Lifestyle' category, RF and SVM show a stronger capability in correctly classifying 'Politics'. SVM outperforms the other two in overall accuracy and particularly in minimizing cross-topic misclassifications. These findings suggest that SVM, with its high precision in classifying high-dimensional data, could be the best-performing model among the three, especially when distinguishing between topics with subtle differences.

Figure 7



The ROC-AUC curves for the three models are displayed above. Both the RF and SVM models exhibit near-perfect ROC curves with AUC scores of 1.00 for the 'Lifestyle' class and 0.99 for the 'Events' and 'Politics' classes. The MNB model's ROC curve for the 'Lifestyle' class is also perfect with an area of 1.00, while the AUC scores for the 'Events' and 'Politics' classes are slightly lower at 0.95.

From these curves, we can infer that all models have a high true positive rate with minimal false positives. The nearly identical performance of the RF and SVM suggests that either of them could be effectively used for this classification task. The final choice may depend on other factors such as model interpretability, training time, or computational resources, which we will consider when deciding on our best model.

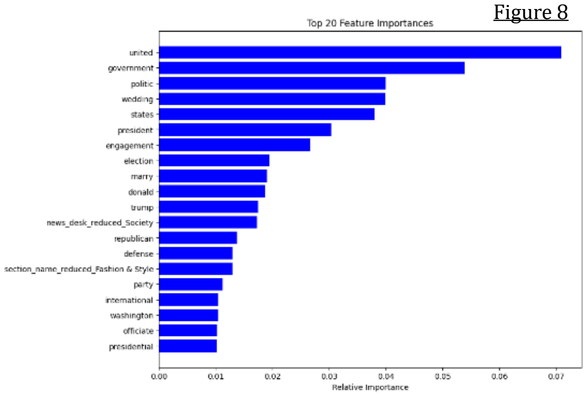
Best Model Justification

Based on the findings, we have selected the default Random Forest (RF) model trained on the original data as our best model. It does not have the highest CV ROC AUC score among all models but achieves the highest score for those trained on the original dataset. Our preference for models trained on the original data is based on four key reasons: First, the models trained on the original data already achieve high metric scores. Second, the original data more accurately represents real-world scenarios, enhancing the practical applicability of the models. Third, models trained on technique-balanced data, such as those utilizing SMOTE, may be prone to overfitting because of their reliance on synthetic data points. Lastly, models trained on original data are easier to maintain and update, eliminating the need for synthetic data generation and simplifying the integration of new, real-world data. While the SVM model shows performance comparable to the RF model, the RF is favored for practical considerations, including computational efficiency and scalability for large datasets. Thus, the RF model stands out as the most suitable choice for our needs, balancing high performance with practicality.

Feature Importance and Ablation Analysis: Random Forest Classifier

For the feature importances analysis, we created a feature importance plot for the top 20 features from our best model. This plot shows the relative importance of each feature in our dataset after one-hot encoding and TF-IDF vectorization. It is evident that the predominant features are derived from the TF-IDF matrix, representing words from the "preprocessed_text" feature in our original dataset before vectorization. Additionally,

Figure 8



we observed that two features not originating from the matrix are included. These supplementary predictors were chosen alongside the main "preprocessed_text" feature to potentially improve model performance. Indeed, the "preprocessed_text" — which we refer to as the main predictor — has played a pivotal role in the classification task. Additionally, the "other features," which we consider helper predictors, have also contributed to enhancing the model's classification capabilities.

In conducting an ablation analysis on our best-performing model, we aimed to understand the contribution of different subsets of features to the model's predictive performance. This was quantified using the mean CV ROC AUC Score across various feature configurations as shown on the below table.

Table 6

Models	**Mean CV ROC AUC Score**
BEST model Trained on original dataset	0.9895735365262885
on dataset with only text feature	0.9891558309570868
on dataset removed top 20 features	0.9634481574686948
on dataset with only text feature and removed top 10 features	0.9656786114014435

Here are key insights summarized below**:**

- Isolating textual features yielded a ROC AUC of 0.9892, nearly matching the 0.9896 of the full feature models, indicating the textual data's strong predictive power.
- Eliminating the top 20 features resulted in a reduced ROC AUC of 0.9634, revealing their significant influence on model accuracy.
- Removing the top 10 features from the text-only dataset led to a ROC AUC of 0.9657, suggesting some redundancy among these features.

Sensitivity Analysis: Random Forest Classifier

We analyzed the sensitivity of the Random Forest classifier by observing the effect of varying hyperparameters and features on the mean cross-validated ROC-AUC score.

First, we tuned the n_estimators parameter of the model. We varied this parameter from 100 to 400, increasing by 50 every time. As we increased the `n_estimators` we found that the ROC-AUC score remained quite high and consistent, generally remaining between 0.989 and 0.990. Next, we analyzed the effect of tuning the max_features parameter. We looked at the ROC-AUC score when max_features was ‘sqrt’,‘log2’, and None. We found that varying this parameter resulted in a mean ROC-AUC score that fell between 0.98 and 0.99. These results suggest that the model is not sensitive to the n_estimators or max_features parameters.

Finally, we tuned the number of features the model was trained on. Since our original dataset had 1064 features, we varied the number of features from 50-1050, adding 50 features each time. We found that initially adding features improves the score, but after around 600 features, adding more features didn't drastically improve the ROC-AUC score, and eventually the score plateaued. This shows that initially the model is sensitive to the number of features it is trained on, but after a certain point, adding more features doesn't significantly improve model performance.

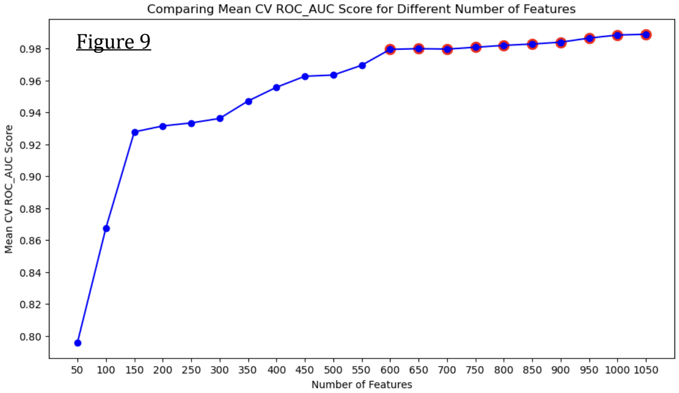


Figure 9

Important Tradeoffs

Analyzing the performance metrics for the Random Forest classifier, we identified some potential tradeoffs. Primarily, we found that in our model evaluation we were able to get slightly better performing models at the expense of the computing speed of the GridSearchCV. Ultimately, as described in the Model Exploration and Evaluation section, we deemed differences in evaluation method as not significant when we determined our best model. Additionally, we found that within the individual topics we can see some tradeoffs present. For example, looking at precision and recall for the articles with the label of "Events", we had a very high precision of 1, but a slightly lower recall of 0.89. However, it is important to note that the precision, recall, accuracy, and f1 scores we obtained were all around 0.89 or higher.

Failure Analysis

We conducted failure analysis of article classification using the Random Forest Classifier on a new dataset of articles from February 2024. We opted to integrate a new dataset despite the model's remarkable performance with the previous dataset as a precaution against potential overfitting. Given the model's unexpectedly high performance on the training dataset, there was a concern that it might have become too tailored to the specific characteristics of that dataset.

Pre-processing the new data, the NMF model identified three topics: "Global Politics", "US Politics", and "Entertainment & Life". The classifier was trained on this data and showed high precision and accuracy, with varying recall scores across categories. The precision scores were 0.95 for "Entertainment & Life" 0.97 for "Global Politics" and 1.00 for "US Politics" while the corresponding recall scores were 1.00, 0.95, and 0.78, respectively. The F1-scores were 0.97, 0.96, and 0.88 for "Entertainment & Life" "Global Politics" and "US Politics" respectively, with support of 133 articles for "Entertainment & Life" 40 articles for "Global Politics" and 27 articles for "US Politics." The overall accuracy of the classifier was 0.96.

The results highlight the classifier's effectiveness in categorizing articles into the identified topics. However, the recall for the "US Politics" category needs improvement. Future work may refine the training process, explore alternative feature engineering techniques, and augment the dataset with diverse samples to enhance the model's performance and reliability.

Discussion

Unsupervised Learning

When performing topic modeling using unsupervised learning methods, an observation that stood out to us was the relative consistency in coherence scores as we varied the hyperparameters. Prior to performing the analyses, we expected a wider range of coherence scores as we varied the number of topics and other hyperparameters. While varying the number of topics with NMF did prove to show some variation in the coherence score, as mentioned in the sensitivity analysis, overall, the scores remained in the range of 0.55 to 0.66.

A significant challenge that we faced in our topic modeling task was hyperparameters tuning of the various unsupervised learning methods. With both LDA and NMF, we experienced a significant reduction in computing speeds when using GridSearchCV or specific looping algorithms to determine the best hyperparameters. To address this, we utilized resources such as Great Lakes, and we revised our hyperparameter selection, focusing primarily on varying the number of topics and comparing the coherence scores.
As both NMF and LDA have many tunable hyperparameters, it is certainly possible that more coherent topics could be generated. As we alluded to above, with more time and computing resources further extension and exploration of our topic modeling could be done by performing more extensive hyperparameter tuning.

Supervised Learning

The analysis revealed that the random forest model, typically considered a baseline or benchmark model, surprisingly outperformed the multinomial naive Bayes and SVM models. This success underscores the importance of empirical testing and validation in machine learning projects rather than assuming superiority based on conventional wisdom. However, a challenge was the class imbalance within the dataset, with the “events” topic containing significantly fewer samples compared to the more abundant “lifestyle” and “politics” topics.

Given more time and resources, an ideal extension would involve setting up a comprehensive pipeline for the machine learning workflow. This would automate tasks from data fetching to topic classification, enhancing the solution's efficiency, reproducibility, and scalability, serving as a precursor for further optimization and refinement.

Ethical Considerations:

For our unsupervised learning approach of topic modeling, there are potential ethical issues that need to be acknowledged. Mainly that of mischaracterizing the topic of articles that are important to the communities of underrepresented groups. While we did not do a deep dive analysis of this particular issue, it is possible that articles that pertain to issues important to underrepresented minority groups are not as prevalent in our dataset. Therefore, using an unsupervised technique like NMF could lead to the topics of those articles being inaccurately labeled. For example, if an informative news article about a minority group contained references to celebrities of that minority group voicing their opinion on an issue, that particular article may be incorrectly labeled as entertainment due to the names of the celebrities being present in the article, when in fact it is important news to that minority group’s community.

Similarly in the text classification supervised learning section, the issue of mischaracterizing articles relating to underrepresented groups may arise as well. If the topic modeling doesn’t properly characterize these articles, classification of these types of articles may be inaccurate as well. For example, if there are very few articles pertaining to issues in the community of an underrepresented group, when classification is done with an article relating to that community it could be done inaccurately, or in a way that misrepresents what the true meaning of the article is. This can be dangerous as articles may be classified as a topic that erroneously diminishes the values of a particular community or paints the issues important to that community in a negative light.

In both supervised and unsupervised learning, we would want to keep these cases in mind and manually check the labels being assigned to these types of articles and ensure that they aren’t being mislabeled and classified. We can also monitor how new instances of these articles are being classified to check for potentially harmful and inaccurate labeling. Additionally, we could monitor how many articles of that type are in our dataset and make an effort to increase the data we have on those articles if they are too low.

Statement of Work:

Qian Fu: Data Collection/Preprocessing, Feature Engineering, Unsupervised Learning evaluation conclusions, Supervised Learning evaluation and conclusions, Supervised Learning Feature Importance

Kevin Deloria: LDA Analysis, Unsupervised Learning Sensitivity Analysis, MNB Analysis, Supervised Learning Failure Analysis, Supervised Learning Discussion, GitHub Notebook Organization/Streamlining

Yash Dave: NMF Analysis, Unsupervised Learning Discussion, Random Forest Classifier Analysis, Supervised Learning Sensitivity Analysis and tradeoffs, Related Works, Ethical Considerations

References

1. Brownlee, J. (2020, August 27). *How to tune the number and size of decision trees with XGBoost in python*. MachineLearningMastery.com. https://machinelearningmastery.com/tune-number-size-decision-trees-xgboost-python/
2. Datascience, R. S. (2023, April 1). *Topic modelling using NMF*. Kaggle. https://www.kaggle.com/code/rockystats/topic-modelling-using-nmf
3. *Gensim.models.coherencemodel*¶. gensim.models.CoherenceModel. (n.d.). https://tedboy.github.io/nlps/generated/generated/gensim.models.CoherenceModel.html
4. Huh, K. (2021, February 13). *Surviving in a random forest with imbalanced datasets*. Medium. https://medium.com/sfu-cspmp/surviving-in-a-random-forest-with-imbalanced-datasets-b98b963d52eb
5. Kardos, M. (2023, August 4). *Unsupervised text classification with topic models and good old human reasoning*. Medium. https://medium.com/@power.up1163/unsupervised-text-classification-with-topic-models-and-good-old-human-reasoning-da297bed7362
6. Manai, H. (2022, January 4). *Machine learning : Text classification of news articles*. Medium. https://hedimanai.medium.com/machine-learning-text-classification-of-news-articles-bd5d70473037
7. Saxena, S. (2023, August 25). *A beginner's Guide to Random Forest hyperparameter tuning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/
8. *Smote#*. SMOTE - Version 0.12.0. (n.d.). https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
9. Srivastava, T. (2023, August 22). *Tuning the parameters of your random forest model*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/
10. *Topic extraction with non-negative matrix factorization and latent Dirichlet allocation*. scikit. (n.d.-a). https://scikit-learn.org/stable/auto_examples/applications/plot_topics_extraction_with_nmf_lda.html
11. *Topic extraction with non-negative matrix factorization and latent Dirichlet allocation*. scikit. (n.d.-b). https://scikit-learn.org/stable/auto_examples/applications/plot_topics_extraction_with_nmf_lda.html
12. Tuncer, C. (2020, November 6). *BBC News Text Classification*. Medium. https://medium.com/analytics-vidhya/bbc-news-text-classification-a1b2a61af903

Appendix 1.1: Data Feature Description and Inclusion Status

Feature Name	Description	Inclusion Status
abstract	A concise overview of the article's content.	Kept for Project Tasks
Web_url	URL of the article	Dropped, Unnecessary Feature
snippet	contains content identical to the abstract	Dropped, Unnecessary Feature
Lead_paragraph	Introductory paragraph of the article.	Kept for Project Tasks
print_section	The section of the print edition	Dropped, Too Many Missing Values
print_page	The page number in the print edition	Dropped, Too Many Missing Values
source	The origin or provider of the article.	Dropped, after EDA
multimedia	Data related to article's multimedia elements.	Dropped, after EDA
headline	Title of the article.	Kept for Project Tasks
keywords	Tags or keywords linked to the article.	Kept for Project Tasks
pub_date	Date and time of publication.	Dropped, after EDA
document_type	Type of document, e.g., article, blog.	Dropped, after EDA
news_desk	Department that produced the article.	Kept for Project Tasks
section_name	Newspaper section containing the article.	Kept for Project Tasks
byline	Author's name and details	Kept for Project Tasks
type_of_material	Nature of content, e.g., News, Obit.	Kept for Project Tasks
_id	Unique identifier for the article.	Dropped, Unnecessary Feature
word_count	Total number of words in the article.	Dropped, after EDA
uri	Unique resource identifier for the article.	Dropped, Unnecessary Feature
subsection_name	Specific subsection within the main section.	Dropped, Too Many Missing Values
slideshow_credits	Credits for slideshow	Dropped, Too Many Missing Values

Appendix 1.2: Final Feature Selection for Both Tasks.

Feature Name	Contents	Usage
Combined_text	Combination of four original features: abstract, lead_paragraph, headline and keywords.	The concatenation of these four features serves as the primary feature for both tasks.
Type_of_material	Predominantly news material, with a ratio of 5:1 compared to other material types.	Predictor in supervised learning task.
News_desk	Principal categories such as “Foreign”, “Metro”, and “Sports”.	Predictor in supervised learning tasks.
Section_name	Primary categories including “U.S.”, “Opinion”, and “New York”.	Predictor in supervised learning task.

author	Author’s first and last name, role, etc.	Predictor in supervised learning task.
Topic_label	Topic label for each article generated from topic modeling model	Target variable in supervised learning task.

Appendix 1.3: Notebooks Used for Each Part and the Functionality of Each Notebook.

Below is the link to our Git repository containing all the notebooks in the "notebooks" folder.

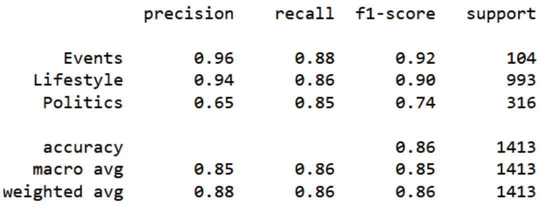
https://github.com/akiwarheit/siads-696-nyt

Category	Notebook Name	Functionality
Part One: Data Preparation	0_1 Data Retrieval	Contains a function designed to retrieve data from the NYT API.
	0_2 Data Cleaning	Performing light cleaning of the raw data
	0_3 Data Exploring	Exploring the features of the dataset to determine their suitability for unsupervised and supervised learning tasks.
	0_4 Data Preprocessing	Preprocessing the selected features for both tasks.
Part Two: Unsupervised Learning	1_0 Feature Engineering Part One	Preparing the primary feature (combined_text) for both tasks.
	1_1 LDA	Exploring and evaluating LDA method
	1_2 NMF	Exploring and evaluating NMF method
	1_3 NMF Sensitivity Analysis	Conducting sensitivity analysis on the best topic modeling model.
Part Three: Supervised Learning	2_0 Feature Engineering Part Two	Additional preprocessing and feature engineering for features used in supervised learning.
	2_1 Stratify Train Test Split	Applying stratified data balancing method.
	2_2 MNB	Exploring MNB method
	2_3 SVM	Exploring and evaluating SVM method
	2_4 RF	Exploring RF method
	2_5 Evaluating MNB	Applying evaluation function to MNB models aligned with SVM evaluation strategy.

	2_6 Evaluating RF	Applying evaluation function to RF models aligned with SVM evaluation strategy.
	2_7 RF Sensitivity Analysis	Conducting sensitivity analysis on the best classification model.
	2_8 Feature Importance Analysis	Conducting feature importance and ablation analysis on the best classification model
	3_0 Feature Engineering Part Three	Preprocessing and feature engineering of newly retrieved data used for failure analysis.
	3_1 Failure Analysis	Utilizing new data to test the best classification model.

Appendix 2.1: Classification Reports for All Classification Models.

Default MNB model trained on original dataset.

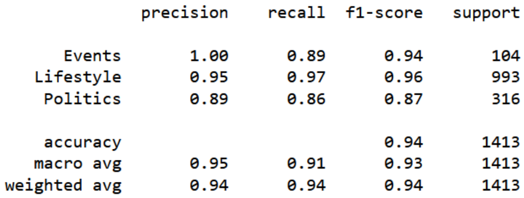


```
              precision    recall  f1-score   support

      Events       0.96      0.88      0.92       104
   Lifestyle       0.94      0.86      0.90       993
    Politics       0.65      0.85      0.74       316

    accuracy                           0.86      1413
   macro avg       0.85      0.86      0.85      1413
weighted avg       0.88      0.86      0.86      1413
```

Default RF model trained on original dataset.

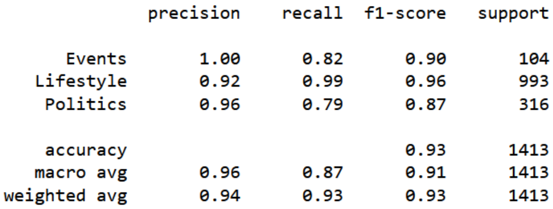


```
              precision    recall  f1-score   support

      Events       1.00      0.89      0.94       104
   Lifestyle       0.95      0.97      0.96       993
    Politics       0.89      0.86      0.87       316

    accuracy                           0.94      1413
   macro avg       0.95      0.91      0.93      1413
weighted avg       0.94      0.94      0.94      1413
```

Default SVM model trained on original dataset.

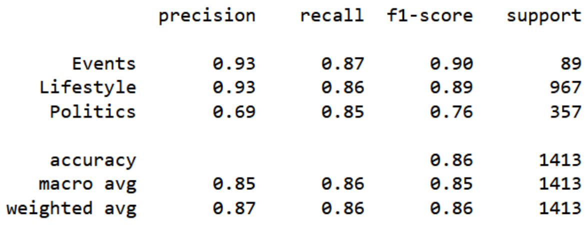


```
              precision    recall  f1-score   support

      Events       1.00      0.82      0.90       104
   Lifestyle       0.92      0.99      0.96       993
    Politics       0.96      0.79      0.87       316

    accuracy                           0.93      1413
   macro avg       0.96      0.87      0.91      1413
weighted avg       0.94      0.93      0.93      1413
```

Default MNB model trained on stratified dataset.

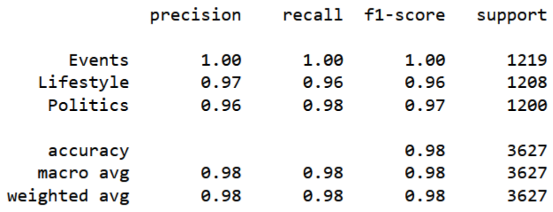


```
              precision    recall  f1-score   support

      Events       0.93      0.87      0.90        89
   Lifestyle       0.93      0.86      0.89       967
    Politics       0.69      0.85      0.76       357

    accuracy                           0.86      1413
   macro avg       0.85      0.86      0.85      1413
weighted avg       0.87      0.86      0.86      1413
```

Default RF model trained on SMOTE-processed dataset.

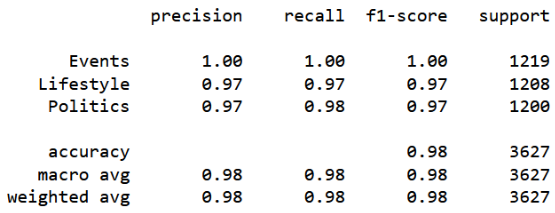


```
              precision    recall  f1-score   support

      Events       1.00      1.00      1.00      1219
   Lifestyle       0.97      0.96      0.96      1208
    Politics       0.96      0.98      0.97      1200

    accuracy                           0.98      3627
   macro avg       0.98      0.98      0.98      3627
weighted avg       0.98      0.98      0.98      3627
```

Default SVM model trained on SMOTE-processed dataset.

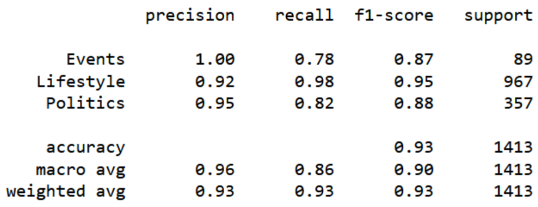


```
              precision    recall  f1-score   support

      Events       1.00      1.00      1.00      1219
   Lifestyle       0.97      0.97      0.97      1208
    Politics       0.97      0.98      0.97      1200

    accuracy                           0.98      3627
   macro avg       0.98      0.98      0.98      3627
weighted avg       0.98      0.98      0.98      3627
```

Default SVM model trained on stratified dataset.

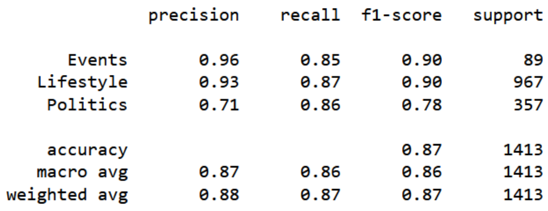


```
              precision    recall  f1-score   support

      Events       1.00      0.78      0.87        89
   Lifestyle       0.92      0.98      0.95       967
    Politics       0.95      0.82      0.88       357

    accuracy                           0.93      1413
   macro avg       0.96      0.86      0.90      1413
weighted avg       0.93      0.93      0.93      1413
```

Tuned MNB model trained on stratified dataset.

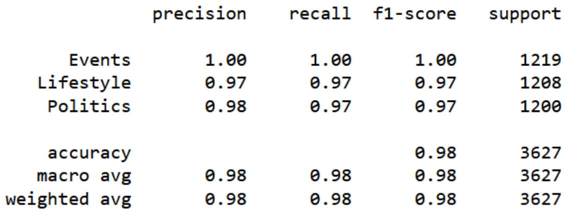


```
              precision    recall  f1-score   support

      Events       0.96      0.85      0.90        89
   Lifestyle       0.93      0.87      0.90       967
    Politics       0.71      0.86      0.78       357

    accuracy                           0.87      1413
   macro avg       0.87      0.86      0.86      1413
weighted avg       0.88      0.87      0.87      1413
```

Tuned RF model trained on SMOTE-processed dataset.



```
              precision    recall  f1-score   support

      Events       1.00      1.00      1.00      1219
   Lifestyle       0.97      0.97      0.97      1208
    Politics       0.98      0.97      0.97      1200

    accuracy                           0.98      3627
   macro avg       0.98      0.98      0.98      3627
weighted avg       0.98      0.98      0.98      3627
```