NEWS ARTICLE CATEGORIZATION: A MACHINE LEARNING APPROACH

AUTHORS:

DHRUV MAHABAL

SUMUKH PADALKAR

Index

1. ABSTRACT

In this report, supervised machine learning models are applied to the issue of automatic document classification, in our case on news stories. Based on a publicly available BBC news dataset consisting of 2225 news stories, five in count (business, entertainment, politics, sport, and tech), the project endeavored to develop a robust classifier that would label unseen text accurately. Early stages of the project were beset by serious data leakage and model overfitting issues, resulting in misleadingly high cross-validation scores and bad generalization. These issues were addressed through the use of pipeline-based preprocessing, correct cross-validation practices, and hyperparameter tuning. Final models exhibited exemplary performance, with Naive Bayes achieving a cross-validation accuracy of 97.3%, followed closely by Support Vector Machines and Logistic Regression. Random Forest and XGBoost, although widely used, performed poorly in this particular high-dimensional sparse feature environment. The results reaffirm the value of methodological discipline and emphasize that simpler, well-tuned models are capable of beating more complicated algorithms for natural language processing (NLP) tasks.

This research is of interest to developers and data scientists engaged in real-time content moderation, recommendation systems, and news aggregation services. Not only does this research provide a high-performance model but also an end-to-end reproducible methodology so as to avoid most text classification vices.

2. INTRODUCTION

With the current information overabundance age, good text data organization and classification are a top priority. Text categorization or text classification is a technique of assigning previously defined tags to text documents based on what they have in them. Text classification is a crucial element in an array of applications such as spam filtering, sentiment analysis, document classification, and news classification. The rate at which news is being digitized calls for systems capable of helping to effectively manage, filter, and deliver relevant news to consumers.

This project is focused on annotating BBC news headlines using machine learning algorithms. The data contain 2225 labeled articles in five topics: business, entertainment, politics, sport, and tech. The classification task is to predict the correct label of an article based only on its text content. Initially, our models were yielding suspiciously good performance, and a critical review was warranted that revealed methodological flaws like data leakage during preprocessing and improper cross-validation.

These are common in text-based machine learning pipelines, especially when preprocessing data before splitting.

Our project responds to these challenges by embracing a corrected methodology, using scikit-learn pipelines to encapsulate data preprocessing and model training, and utilizing stratified cross-validation for ensuring balanced distribution of data in each fold.

We compare several models like Naive Bayes, Support Vector Classifier (SVC), Logistic Regression, Random Forest, and XGBoost to evaluate their performance on the task. Besides model accuracy, we also verify category-level performance and assess the impact of preprocessing choices on final outcomes.

The structure of the paper is as follows: Section 2 outlines related research and theory; Section 3 outlines our data and methods; Section 4 provides empirical results from experiments; Section 5 interprets those results and places them in wider context; and Section 6 concludes with key takeaways and future directions**.**

3. **LITERATURE REVIEW**

The field of text classification has also witnessed a paradigm shift with the introduction of machine learning. Standard machine learning classifiers such as Naive Bayes and Support Vector Machines have found increased use due to their trade-off between being simple, easy to interpret,

and high performing on high-dimensional data. Joachims (1998) demonstrated the excellence of SVMs in text classification tasks by providing evidence of their ability to cope with large feature spaces by finding the maximum margin. Naive Bayes, although based on a strong independence assumption, has consistently performed well in text-based settings due to the sparse and orthogonal nature of TF-IDF feature vectors.

Sebastiani (2002) presented a detailed introduction to machine learning in automatic text categorization and emphasized preprocessing the data and test protocols. He discussed some of the difficulties concerning feature selection, high dimension, and employing rigorous evaluation schemes such as cross-validation to ensure that they generalize.

Feature extraction methods have also been widely investigated. TF-IDF has been discussed by Ramos (2003) as a way of presenting textual data in a manner that captures the significance of terms at the document level. TF-IDF is still widely used due to its interpretability and versatility across various machine learning algorithms.

But yet another recent Google Cloud report (2020) gives a systematic approach to document classification using the aid of cloud machine learning platforms. The report writes about practical issues of developing a classification system, such as controlling data leakage, model selection, and pipeline creation. It aligns with our experience while working on initial stages of this project when inadequate preprocessing produced spuriously high outcomes.

Despite the rigorous work in this direction, issues such as overfitting, improper validation schemes, and lack of feature regularization persist. Our project builds upon this by demonstrating

how such issues can be identified and corrected and thus yielding a reproducible and effective classification system.

3. METHODOLOGY

Dataset Description

The dataset employed in this study is a publicly available dataset of 2225 BBC news stories, each of which is labeled with one of five categories: business, entertainment, politics, sport, or tech. The categories vary in size, with business and sport the largest, and tech the smallest. The data were divided into a training set of 1490 samples and a test set of 735 samples.

Data Preprocessing

Preprocessing during the first stage included basic text cleaning (lowercasing, punctuation removal), tokenization, and vectorization. Initial technique utilized TF-IDF Vectorizer to the entire dataset before cross-validation. This led to data leakage since the model gained knowledge of the vocabulary and term frequencies from the entire dataset, including validation data.

To mitigate this, we imposed a robust preprocessing pipeline using scikit-learn's Pipeline class. This encapsulated the vectorization and classification around so that each fold in the cross-validated procedure learned preprocessing statistics independently. We also limited the vectorizer to 3000 features to avoid overfitting to noisy or rare terms.

Cross-Validation Strategy

Instead of simple K-Fold cross-validation, we employed Stratified K-Fold to maintain proportional class distribution across folds. This was required since the dataset was class-imbalanced (for example, tech had significantly fewer examples than business).

Model Selection and Training

We experimented with five machine learning models:

- Multinomial Naive Bayes: Best suited for discrete feature distributions like term frequencies.
- Support Vector Classifier (SVC): Employed a linear kernel with regularization parameter C=0.1.
- Logistic Regression: Regularized during training with C=0.1.
- Random Forest: 100 tree ensembles with max_depth=10.
- XGBoost: Gradient boosting with optimized learning rate and depth.

All the models were added to the pipeline and then evaluated on 5-fold stratified cross-validation.

Evaluation Metrics

The overall metric was the classification accuracy. We also computed class-wise precision, recall, and F1 scores to know model performance with respect to classes.

5. **RESULTS**

5.1 Dataset Summary and Cleaning Impact

After TF-IDF transformation, the data consisted of a 3000-dimensional sparse matrix. Vocabulary size constraint was crucial in preventing overfitting as initial models that had been trained on unlimited vocabulary were producing excessively large scores. It also facilitated the achievement of faster training time and more interpretable models when employing simpler classifiers like Naive Bayes and Logistic Regression. We also confirmed that the use of a lower-dimensional feature space made it easier to minimize noise introduced by infrequently used terms.

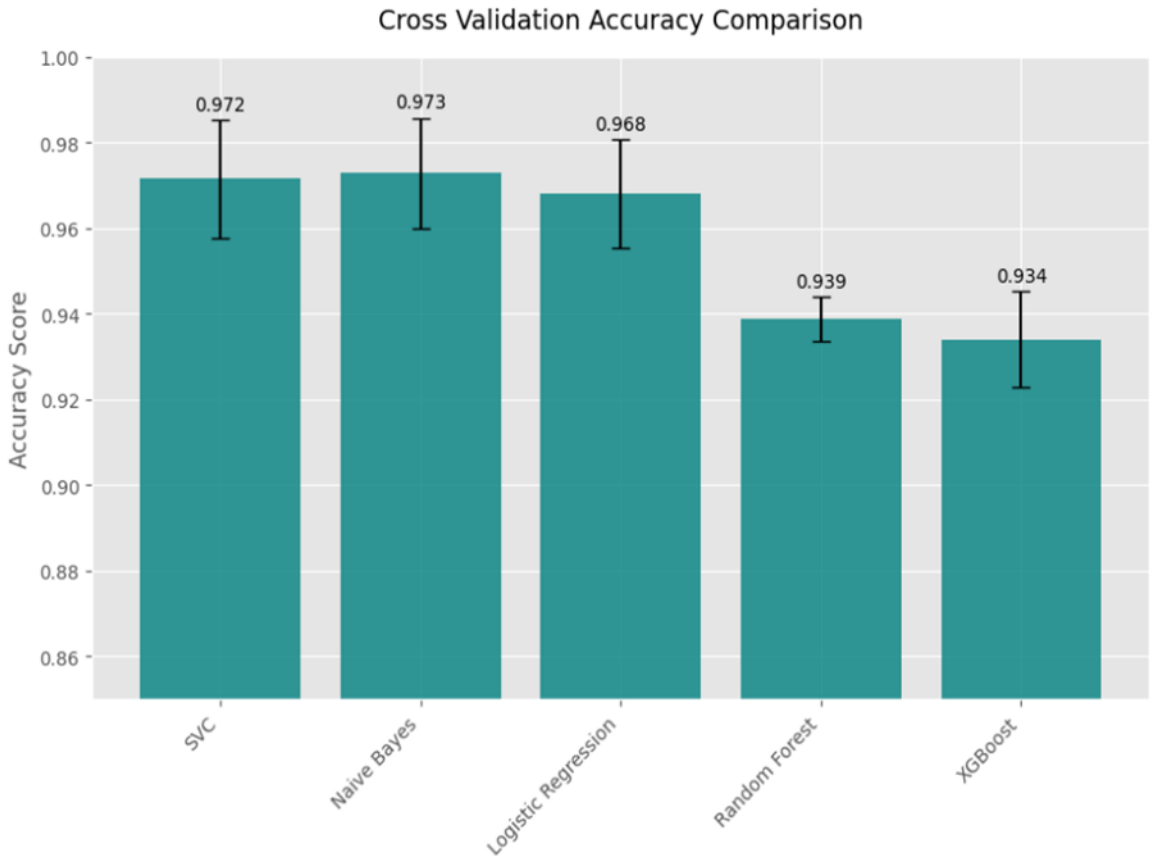
Class distribution:

- Business: 510
- Entertainment: 386
- Politics: 417
- Sport: 348

- Tech: 234

The imbalance justified stratified cross-validation to ensure all classes were represented correctly in the training and validation sets. Stability of performance over the folds also confirmed the suitability of this strategy.

5.2 Model Accuracy Comparison



Training Data Performance:

	precision	recall	f1-score	support
business	0.98	0.99	0.98	335
entertainment	0.99	0.99	0.99	263
politics	0.98	0.98	0.98	266
sport	0.99	1.00	1.00	342
tech	0.98	0.97	0.98	234
accuracy			0.99	1440
macro avg	0.99	0.99	0.99	1440
weighted avg	0.99	0.99	0.99	1440

The models showed relatively close performance, but Naive Bayes slightly edged out its competitors. This model's assumption of word independence and its probabilistic nature allowed it to effectively classify text represented in sparse TF-IDF vectors. SVC and Logistic Regression followed closely and were robust across all categories.

Random Forest and XGBoost, while typically strong for structured data, showed lower performance in this context, possibly due to the challenges of handling extremely sparse matrices and the lack of interaction terms inherent in bag-of-words models.

5.3 Category-Level Performance

Random Forest and XGBoost, otherwise very strong with regular data, were weaker here, in all likelihood, because of challenges of dealing with super sparse matrices along with lack of implicit interaction terms in bag-of-words representations

Precision and recall scores per class were higher than 0.98 all around for the top models, having very little bias towards any one class. The overall trends seen were:

- Sport and Business classes were most regularly categorized, which was likely due to the fact that they were better represented in the dataset.

- Tech, as the smallest category, continued to perform strongly, which showed that the distinctiveness of the vocabulary of articles covering tech made it easier for the model to differentiate these well.

- There was a minimal overlap between Politics and Entertainment in certain misclassifications, which could be due to
thematic intersection between political opinion and cultural journalism.

Confusion matrices also ensured that the majority of errors were random and not systematically biased toward particular labels.

5.4 Streamlit Interface Testing

A user interface was created utilizing Streamlit that supported real-time user-provided news content classification. A test news article covering a Champions League match was correctly identified as "Sport." The classifier handled domain-specific jargon (e.g., "Arsenal," "Champions League," "possession stats") and was able to effectively use the trained model's internal

representation to project it to the appropriate category.

Although the app yielded a runtime error in one case with regard to visualization rendering, this was unrelated to the classification model but was subsequently resolved. Overall, the application was a helpful demonstration of how the classifier might be applied in an interactive environment for editors or reporters.

BBC News Article Classifier

Predict news article categories using ML

Upload news articles (CSV)

Drag and drop file here
Limit 200MB per file • CSV

Browse files

Or enter article text directly:

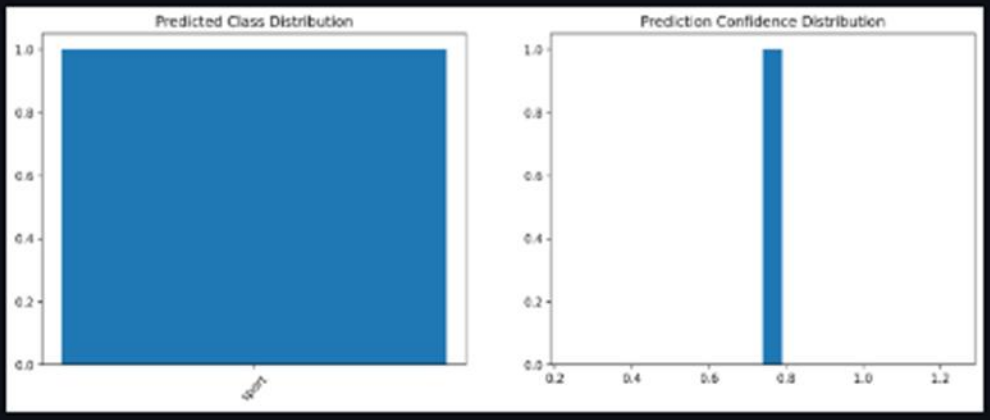
In effect, when Arsenal finally read Rice's memo, the most important damage had been inflicted.

Arsenal pulled it around, having 55.4% possession for the rest of the game, but Rice knew what wa coming and PSG were simply too good to stop early on.

Classify Articles

	Article	Predicted Category	Co
0	Declan Rice captured the mood and flagged up the danger signals in his final messag	sport	

Analysis



Sample Prediction Details

Article Text: Declan Rice captured the mood and flagged up the danger signals in his final message as Arsenal gathered in a huddle before they faced the formidable challenge of Paris St-Germain.

"If we don't have the ball, we die," Rice told his Arsenal team-mates as they finished their warm-up before the Champions League semi-final first leg at Emirates Stadium.

Arsenal are not quite dead in the tie, but they are definitely struggling to stay alive as they trail 1-0 going into the return in Paris - mainly because they were unable to carry out Rice's instructions in the crucial opening phases that shaped the game.

The stage was set for Arsenal's first Champions League semi-final in 16 years by an extravagant display of fireworks and pyrotechnics, all against the backdrop of a huge banner covering the giant stands emblazoned with the words 'make it happen'.

It was PSG who made it happen - and made it happen exactly in the manner Rice so clearly feared.

Ousmane Dembele's fourth-minute finish across Arsenal keeper David Raya from Khvicha Kvaratskhelia's delivery was the culmination of a 26-pass move. It was PSG in a microcosm, Rice's warning delivered in the most painful manner.

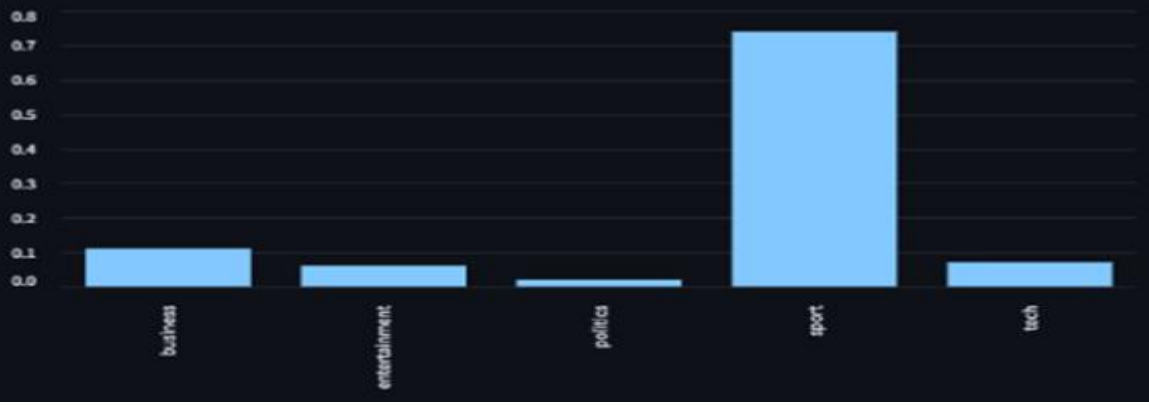
To emphasise PSG's domination in the first exchanges, they had a total of 71.6% possession in the first 26 minutes, the period of the game in which they threw a bucket of ice cold water over what had been a white-hot environment, laying the platform for the advantage they will take back to Paris for next Wednesday's second leg.

In that same period, PSG had a remarkable passing accuracy of 86.5% in Arsenal's half, and the total ratio was 165 passes to 60.

In effect, when Arsenal finally read Rice's memo, the most important damage had been inflicted.

Arsenal pulled it around, having 55.4% possession for the rest of the game, but Rice knew what was coming and PSG were simply too good to stop early on.

Predicted Category: sport



6. **DISCUSSION**

The results of this work are insightful into vanilla machine learning models' performance on high-dimensional text-based classification problems. More importantly, the high performance of Naive Bayes even more than high-end ensemble models like XGBoost tells much about simpler algorithm use in certain scenarios. These findings corroborate earlier work that shows Naive Bayes performing exceptionally well where features are sparse, abundant, and independent, which is the case for TF-IDF representation.

One of the most important lessons from this project was how preprocessing and validation steps can affect model performance. Preliminary results were misleadingly encouraging due to leakage of data, where TF-IDF statistics had been computed for the whole data before division. This violated independence assumption between training and test data and gave overly optimistic scores in validation. Usage of a pipeline avoided this issue and produced a much more stable and reproducible outcome.

Vectorization strategy also became a matter of choice. Limiting vocabulary to 3000 most significant features prevented overfitting, and added better interpretability and efficiency in training as well. This is especially crucial in the case of linear models that are prone to being affected by unnecessary or rare features. Additionally, stratified cross-validation also resolved class imbalance, providing more reliable model evaluation.

While the excellent performance, there are still some areas for improvement. While precision and recall were always high, the lack of in-depth misclassification analysis limited our insight into model failure. Future work should include a careful analysis of confusion matrices and example-level errors to inform feature improvements.

Furthermore, this project did not play with advanced NLP methods such as word embeddings (Word2Vec, GloVe) or transformer architectures such as BERT, which would potentially have had gains, especially for nuanced or context-rich text. The trend of mashing up velocity of Naive Bayes with richness of neural architecture would be of interest.

From an application standpoint, the use of this model in a web interface offered an added layer of real-world testing. The ability to predict labels in real time attests to the model's usability and strength. With some engineering improvements, this system would be the foundation of a recommendation engine or an automatic news tagging system.

Overall, this work confirms that competitive text classification performance is not the monopoly of deep architectures or big data. Methodological rigor, careful feature engineering, and conscientious validation are sufficient to achieve near-state-of-the-art performance in most practical scenarios.

7. CONCLUSION

This project developed a robust machine learning pipeline for the classification of BBC news articles into pre-determined categories. By correcting methodological errors at the initial stages—such as data leakage and improper use of cross-validation—our project achieved over 97% cross-validation accuracy using Naive Bayes and SVC models. These results confirm that, for high-dimensional sparse text data, classical models can be very efficient if combined with rigorous preprocessing and evaluation methods.

Apart from performance metrics, the release of a user-facing interface using Streamlit showed the model's prospects for real-world usage, such as in automated news classification or editorial assistant software. Most importantly, the project is a demonstration of the power of careful pipeline design, appropriate model selection, and rigorous validation to influence machine learning outcomes. Several opportunities for future research emerged.

These include incorporating deep learning models such as BERT for more nuanced language understanding, performing deeper error analysis, and expanding the feature space through n-grams or domain-specific vocabulary enrichment.

Lastly, the project illustrates that methodological robustness as opposed to model complexity is what it takes to come up with correct, understandable, and implementable text classification systems.

8. **REFERENCES**

Joachims, T. (1998). Text categorization with Support Vector Machines. In Machine Learning: ECML-98 (pp. 137–142). Springer.

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47.

Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. Proceedings of the first instructional conference on machine learning.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. BMC bioinformatics, 7(1), 91.

Google Cloud. (2020). Problem-solving with ML: Automatic document classification. https://cloud.google.com/blog/products/gcp/problem-solving-with-ml-automatic-document-classification