WESTERN MICHIGAN UNIVERSITY

STAT-5870 Big Data Analysis Using Python

Text Classification of BBC News Articles

FALL 2022

INSTRUCTOR: Dr. Kevin H. Lee

AUTHORS: Gnana Deepak Madduri, Asif Irfanullah Masum, Ifrat Zaman

I. INTRODUCTION

News reporting has been responsible for educating the general population on a plethora of topics for many years. Reporting news in the form of newspapers is the oldest form of news media. Even a decade ago, it was common to get newspapers delivered at your doorstep every morning to stay on top of everything happening around the globe. However, newspapers are a thing of the past now. With rapid technological advancements, news can be consumed directly from smart devices.

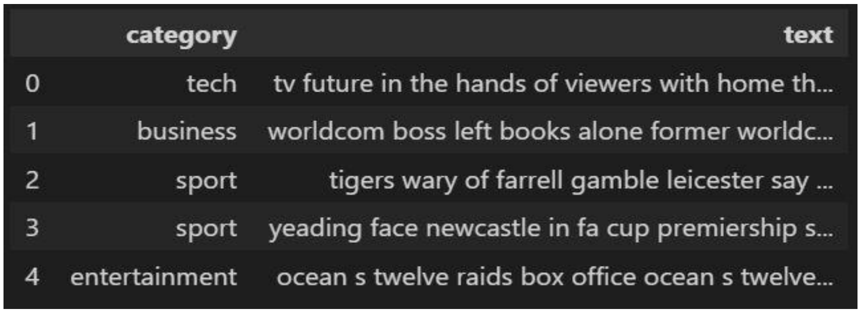
This paradigm shift to digital news comes with its own set of challenges. One such challenge is the engagement and retention of users to a digital news platform. User engagement and retention can be increased by targeting audiences to their specific genre of interest. This can be achieved using Machine Learning (ML) methods and techniques to parse through news articles and categorize each article to a specific news genre. Classifying news articles based on their genre will greatly improve user experience in navigating through a digital news platform. This improved user experience will eventually yield better user engagement and retention.

PROJECT OBJECTIVE

The objective of this project is to create an application using Natural Language Processing (NLP) techniques that can parse through the texts of a dataset consisting of news articles and categorize each article to its specific news genre. The first half of the project is dedicated to identifying, training, and choosing the best ML model with the best accuracy, precision, and recall scores to implement into the application. The application will allow a user to import a news articles dataset that the trained ML model will test and predict the news genre for each article.

DATASET DESCRIPTION

The following dataset was collected from Kaggle [1]. It consists of 2225 numbers of observations with 2 variables: *category* and *text*. The first variable ("*category*") is a classification variable with the following categories: business, entertainment, politics, sports, and tech. The second variable ("*text*") is a string variable that contains the content of a news article from the "*category*" genre.



	category	text
0	tech	tv future in the hands of viewers with home th...
1	business	worldcom boss left books alone former worldc...
2	sport	tigers wary of farrell gamble leicester say ...
3	sport	yeading face newcastle in fa cup premiership s...
4	entertainment	ocean s twelve raids box office ocean s twelve...

Figure 1: Snapshot of the first 5 observations in the dataset

II. METHODOLOGY

This section will explain the data cleaning, preprocessing, exploration, implementation and evaluation of ML models that predict news article genre using the dataset explained above.

ARCHITECTURE & DIAGRAM

In order to choose the best ML model for the application, it is necessary to follow the traditional process of data cleaning, exploration, implementing various ML models, and ML models evaluations. The following diagram is a summary of the methodology used to choose the best ML model for our application.

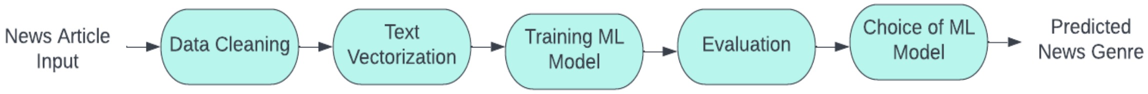


Figure 2: Summary of methodology to choose ML model

DATA EXPLORATION

The best part of the dataset chosen for this project is that it contained no missing values. This makes data exploration more meaningful and prevents ML models from returning biased estimates. Since the dataset only contains the news articles and their genres, it is useful to count the number of articles in each of these genres and visualize it using a box plot.

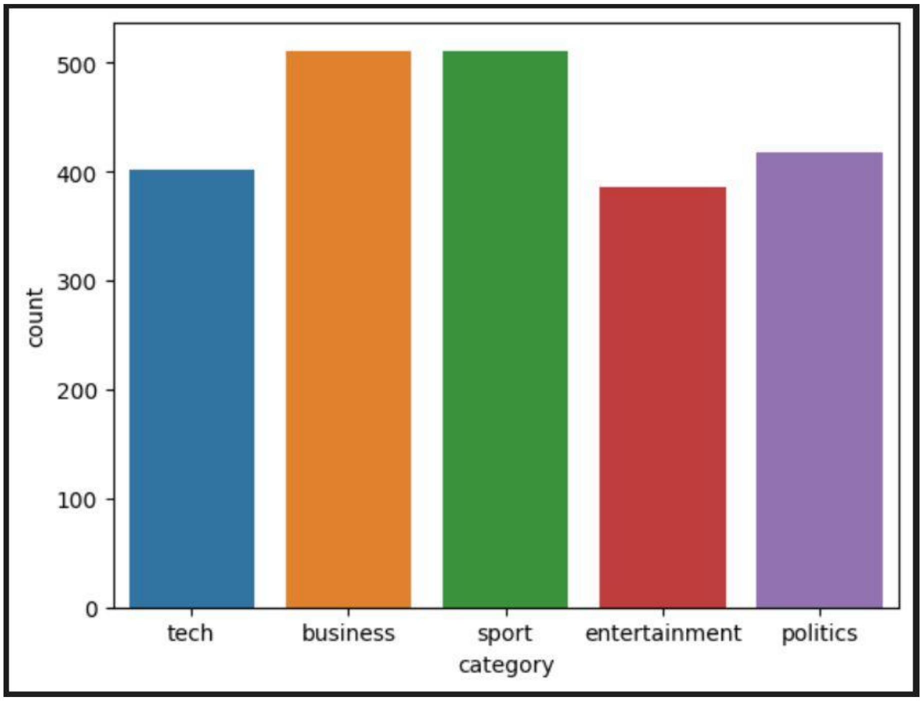
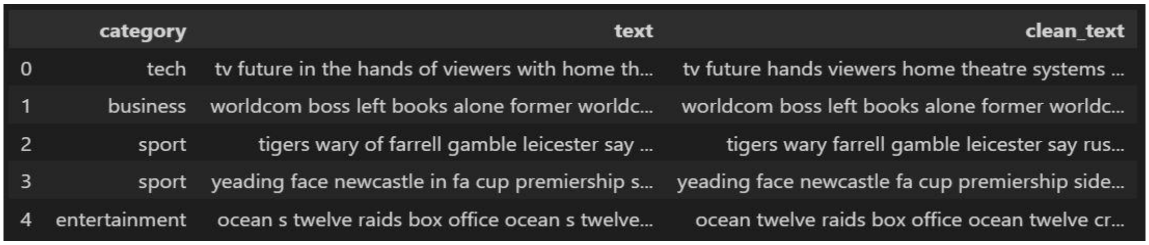


Figure 3: Count of news articles in each genre

DATA CLEANING & PREPROCESSING

The following data cleaning and preprocessing steps were taken in the project:

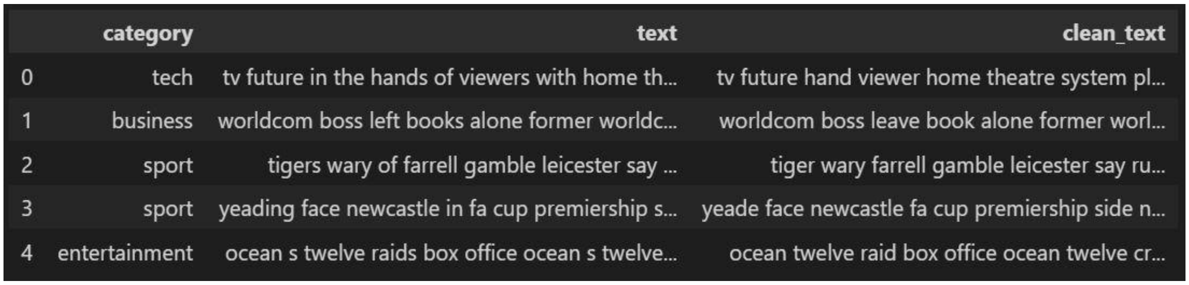
- Lower Case - It is a part of the data cleaning process to change the input text to lowercase in order to have consistency
- Remove Punctuations and Stopwords: Removing punctuations and stopwords that may be present in the input text to filter the unclean data using *NLTK*



	category	text	clean_text
0	tech	tv future in the hands of viewers with home th...	tv future hands viewers home theatre systems ...
1	business	worldcom boss left books alone former worldc...	worldcom boss left books alone former worldc...
2	sport	tigers wary of farrell gamble leicester say ...	tigers wary farrell gamble leicester say rus...
3	sport	yeading face newcastle in fa cup premiership s...	yeading face newcastle fa cup premiership side...
4	entertainment	ocean s twelve raids box office ocean s twelve...	ocean twelve raids box office ocean twelve cr...

Figure 4: Input Text vs Clean Text after removing Stopwords

- Lemmatization: Employing lemmatization using *SPACY* in order to remove inflectional endings only and to return the base or dictionary form of a word



	category	text	clean_text
0	tech	tv future in the hands of viewers with home th...	tv future hand viewer home theatre system pl...
1	business	worldcom boss left books alone former worldc...	worldcom boss leave book alone former worl...
2	sport	tigers wary of farrell gamble leicester say ...	tiger wary farrell gamble leicester say ru...
3	sport	yeading face newcastle in fa cup premiership s...	yeade face newcastle fa cup premiership side n...
4	entertainment	ocean s twelve raids box office ocean s twelve...	ocean twelve raid box office ocean twelve cr...

Figure 5: Text vs Clean Text after Lemmatization

- Vectorization: Using the *Count Vectorizer* and *TF-IDF Vectorizer* to convert input data from its raw format (i.e. text) into vectors of real numbers which is the format that ML models support.

TRAINING ML MODELS AND EVALUATION

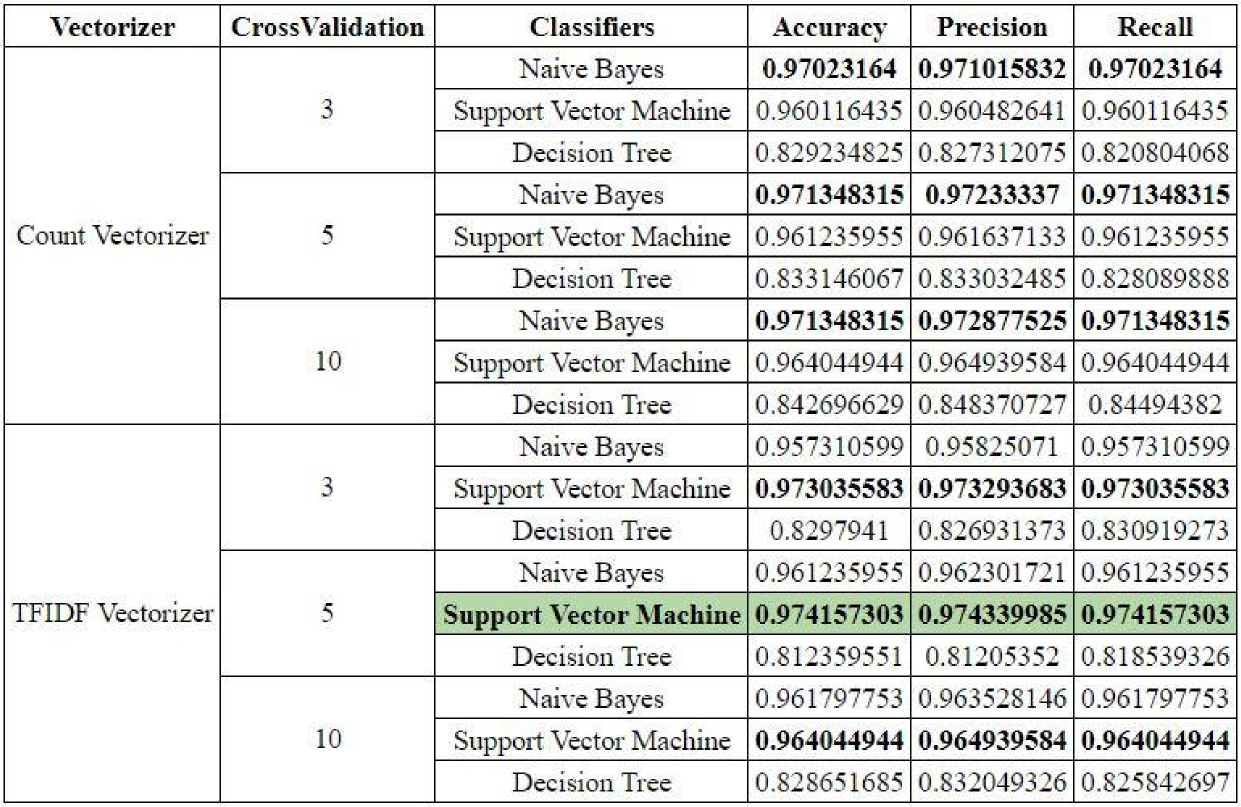
It is now time to train several ML models to evaluate which one works best. The following are the ML models used:

1) *Multinomial Naive Bayes*: One of the popular ML models used for text classification applications. The best feature of Multinomial NB is that it is a collection of algorithms each of which follows the principle that each feature being classified is not related to any other features.
2) *Support Vector Machine*: Another popular text classification ML model. It is an algorithm that divides vectors belonging to the same group.

3) *Decision Trees*: A simple supervised learning algorithm that divides data points according to certain parameters.

CHOICE OF ML MODEL AND EVALUATION

The ML models discussed in the previous section were trained and tested using both Count Vectorizer and TF-IDF methods. Additionally, each ML model was cross-validated using 3, 5, and 10 folds for each vectorizer. Therefore, a total of 18 iterations were implemented and tested (3 ML models, 2 vectorizers, and 3 cross-validations) to identify the best performing ML model for news article genre predictions. The following table summarizes the outputs obtained from the trained ML model predictions for all iterations:

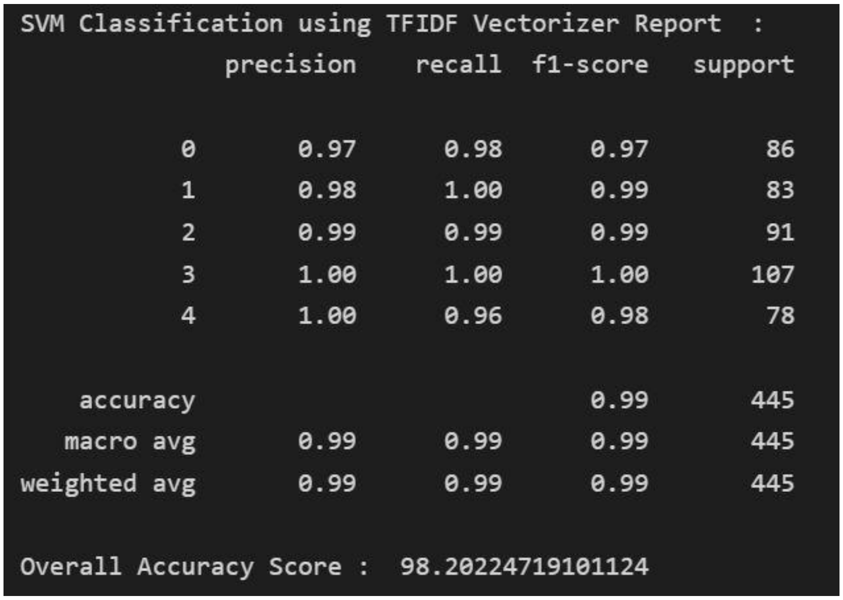


Vectorizer	CrossValidation	Classifiers	Accuracy	Precision	Recall
Count Vectorizer	3	Naive Bayes	**0.97023164**	**0.971015832**	**0.97023164**
		Support Vector Machine	0.960116435	0.960482641	0.960116435
		Decision Tree	0.829234825	0.827312075	0.820804068
	5	Naive Bayes	**0.971348315**	**0.97233337**	**0.971348315**
		Support Vector Machine	0.961235955	0.961637133	0.961235955
		Decision Tree	0.833146067	0.833032485	0.828089888
	10	Naive Bayes	**0.971348315**	**0.972877525**	**0.971348315**
		Support Vector Machine	0.964044944	0.964939584	0.964044944
		Decision Tree	0.842696629	0.848370727	0.84494382
TFIDF Vectorizer	3	Naive Bayes	0.957310599	0.95825071	0.957310599
		Support Vector Machine	**0.973035583**	**0.973293683**	**0.973035583**
		Decision Tree	0.8297941	0.826931373	0.830919273
	5	Naive Bayes	0.961235955	0.962301721	0.961235955
		Support Vector Machine	**0.974157303**	**0.974339985**	**0.974157303**
		Decision Tree	0.812359551	0.81205352	0.818539326
	10	Naive Bayes	0.961797753	0.963528146	0.961797753
		Support Vector Machine	**0.964044944**	**0.964939584**	**0.964044944**
		Decision Tree	0.828651685	0.832049326	0.825842697

Table 1: Summary score outputs of all 18 iterations

As indicated above,, the SVM model (kernel=linear) using TF-IDF Vectorizer and CV = 5 yielded the best accuracy **97.42%**. Furthermore, using SVM model with TF-IDF Vectorizer, we train the model using training data from train-test split to get the highest accuracy of **98.20%** for the model.

The following is the summary scores for this specific iteration:



```
SVM Classification using TFIDF Vectorizer Report  :
              precision    recall  f1-score   support

           0       0.97      0.98      0.97        86
           1       0.98      1.00      0.99        83
           2       0.99      0.99      0.99        91
           3       1.00      1.00      1.00       107
           4       1.00      0.96      0.98        78

    accuracy                           0.99       445
   macro avg       0.99      0.99      0.99       445
weighted avg       0.99      0.99      0.99       445

Overall Accuracy Score :  98.20224719101124
```

Figure 6: Summary report of SVM model using TF-IDF Vectorizer and CV = 5

III. APPLICATION

From the previous section, the Support Vector Machine was identified as the best performing model with an accuracy of **98.20%** in predicting the genre of an input text. This robust model can be used in an application to categorize an input text and organize the articles in a system.

Classifying news stories using this methodology would enable organizations to carry out large-scale data backfills, which would give the data warehouse a consistent structure, ideally free from bias and human mistakes. The workload of the journalists, who are presently marking their stories manually, would be decreased. Subsequently, having proper categorization is essential for digital news applications to allow users to view content curated to their specific interest. Therefore, the use and implementation of this project is promising.

IV. CONCLUSION

Text Categorization is an important aspect in Natural Language Processing. Due to pervasive availability and usage of Digital Media, text based categorization may play a key role in developing intelligent applications. In this report, we used BBC News dataset and initial benchmark results with multiple classifiers and multiple folds of experiments. We designed an effective strategy to clean the text in the dataset to remove stopwords and punctuation. In addition to that, we also lemmatized the text and transformed the cleaned text to feature vector using TF-IDF Vectorizer for classification purposes. We have obtained the highest performance of 98.20% accuracy with an SVM classifier equipped with a linear kernel for the dataset. In future, we would like to apply the developed method on other datasets as well and analyze in terms of variations in cluster size, orientations, local and global feature combination and deep features.

V. REFERENCES

[1] Singh, J. (2019) *BBC Dataset*, *Kaggle*. *https://www.kaggle.com/datasets/sainijagjit/bbc-dataset*
[2] *Multinomial naive Bayes explained: Function, Advantages & disadvantages, applications in 2023* (2022) *upGrad blog.https://www.upgrad.com/blog/multinomial-naive-bayes-explained/*
[3] *What is text classification? MonkeyLearn*. Available at: *https://monkeylearn.com/what-is-text-classification/*.
[4] *Text classification using support vector machines (SVM)* (no date) *MonkeyLearn*. Available at: *https://monkeylearn.com/text-classification-support-vector-machines-svm/*
[5] Koenig, R. (2019) *NLP for Beginners: Cleaning & Preprocessing Text Data*, *Medium*. Towards Data Science. *https://towardsdatascience.com/nlp-for-beginners-cleaning-preprocessing-text-data-ae8e306bef0f*