Using machine learning to classify news articles

We use Natural language processing techniques to develop a model that classifies news articles into one of the following five types: sports, politics, technology, business or entertainment. To build this model, we use a data set of BBC news articles. Our model proves to be very accurate.

Introduction

Machine learning is a field of computer science where data is used to develop computational models to make predictions and decisions. Machine learning is applied successfully in numerous fields.

Consider figure 1 below. This is not the exact dataset that is considered in this article, but it will help us illustrate the different concepts we will discuss. The problem we consider in this article belongs to the class known as a supervised learning problem. Each row in that table contains the text of an article, in the first column, and the category it belongs to, in the second column. The text is called the feature of the article. The category is called the label of the article. The text and the category together is called an example. Note that we know both the features and the label of each example in the table. The goal of machine learning is to develop using a data set such as Table 1, i.e. a collection of examples, where both the features and labels of the examples are known. The goal in developing such a model is to later use the model to predict the label of new examples from their features. In other words, the model takes as input the features of an example (the text of an article in our case) and gives as output a prediction of the label of the example (the category of the article in our case). We refer the reader to [1, 2, 3] for more details on Machine learning.

| ArticleID | Text | Category |
| --- | --- | --- |
| 101 | ‘Apple stock soars to new highs’ | business |
| 102 | ‘India elects new president’ | politics |
| 103 | ‘New apple M1 chip has been unveiled’ | tech |
| 104 | ‘Warriors win NBA championship’ | sports |
| 105 | ‘Kanye west and kim kardashian divorce’ | entertainment |

Figure 1

A common practice of machine learning - particularly in supervised learning problems - which we had used during this project is splitting up the data into training and validation sets. The data had been split ⅔ for inputting in order to train the model and ⅓ of the data was used to validate and check the accuracy of the model. This is done to evaluate the true accuracy of the computational model to ensure that it has not overfit the data and thus allows us to check if the hyperparameters or parameters must be changed.

During the rest of this article the techniques we have used to develop the computational model that was able to predict categories will be explained and the testing used to evaluate the model such as a confusion matrix.

Preprocessing the data

Note that the labels have string and not numerical values in figure 1 above. Thus added a fourth column called CategoryID, the values in this column are a number between 0-4 to act as a numerical classification for each category that the model can interpret. See figure 2 below. We then must carry out pre processing on the features (Text).

| ArticleID | Text | Category | CategoryID |
| --- | --- | --- | --- |
| 101 | ‘Apple stock soars to new highs’ | business | 0 |
| 102 | ‘India elects new president’ | politics | 2 |
| 103 | ‘New apple M1 chip has been unveiled’ | tech | 1 |
| 104 | ‘Warriors win NBA championship’ | sports | 3 |
| 105 | ‘Kanye west and kim kardashian divorce’ | entertainment | 4 |

Figure 2

We carried out vectorization on the text features in order so that they can be inputted into the model. Vectorization is a method in NLP in which words and phrases in text are mapped to a corresponding vector of numbers that can be inputted into the model.

We had used TF-IDF vectorization; this is where the weight(vector) of the word is created by multiplying the term frequency - the number of times a word or phrase appears in the text - by the inverse document frequency. The document frequency is the number of documents in the dataset containing the word/phrase, the inverse document frequency can therefore be thought of as the weight of the term as words that appear in fewer documents have a higher value due to them being more rare. The idf is calculated by:

IDF = log(n/df)

Where n is the number of documents and DF is the document frequency of the word. The TF-IDF of the term is then calculated by multiplying the two values (term frequency and IDF). The

The parameters that we had set with TfIDf to optimize the solution were to consider both unigrams and bigrams, using stop words (sklearn’s list of english stop words), ignoring terms with a df frequency of lower than 5 - as these words would be too anomalous to base predictions off of. Finally, we had also set term frequency to sublinear which is where the tf is set to 1 + logtf if tf > 0, otherwise tf will be set to 0. This is because the number of occurrences of the term in the document does not necessarily mean that the term is more significant. The resulting table of the features after the vectorization can be seen below in figure 4. Note this is not the exact table we used yet a mere visual representation.

|  | ‘apple’ | ‘bbc’ | ‘dollar’ | ‘stock’ |
| --- | --- | --- | --- | --- |
| 0 | 0.00000000 | 0.0000000 | 2.45359259 | 1.3359530 |
| 1 | 2.39450 | 0.000000 | 1.358593 | 1.385920 |
| 2 | 0.000000 | 3.458302 | 1.48329 | 0.000000 |
| 3 | 0.00000 | 0.0000 | 3.38502 | 0.0000 |

Figure 3

As seen on figure 4 each word has a corresponding vector in each document (labeled rows 0-4). These features are inputted into the model as the model will base predictions on the known categories of each document and their word vectors.

Multi Classification Problem

Each document/text belongs to one of 5 categories, which means that the model must predict between one of 5 values making this a multi classification problem.

We convert the categoryID column into an array of 5 numbers with each number set to 0 except the corresponding category of the label which is set to 1. Allowing the model to categorize each example based on this. This process is known as one hot encoding.

Using one hot encoding the model predicts the label of an example by creating an array of probabilities for the category which would look as follows for example :

yi = [0.01, 0.02, 0.9, 0.4, 0.3].

With the value in the 2nd index being highest the model will predict the example to belong to this category and thus will predict this example to be in the category of politics.

Logistic regression

Logistic regression is a machine learning technique that can be used to develop models for multi classification and is the specific technique we use in this article.

Logistic regression uses the sigmoid function which is :

$σ\left(x\right)=\frac{1}{1+e^{-x}}.$

The graph of it can be seen below in figure 5



Figure 5

It is important to note that for all x the sigmoid function is between 0-1 and $σ$(0) is the midpoint of the function at 0.5. In the terms of our problem if the sigmoid for a particular label (category) for an example is above 0.5 it will predict it to be that specific label.

Although, there are multiple features that are used to calculate our prediction $\hat{y}$ In our specific problem for the sake of simplicity we will assume that there are only 2 features - x1 and x2. Given this, the functional form of our prediction $\hat{y}$ is assumed to be

$\hat{y}= σ\left(w\_{1}x\_{1}+w\_{2}x\_{2}+b\right),$

W1, w2 and b are parameters that have been chosen. These parameters are chosen in practice by software that chooses the parameters that produce the lowest possible mean categorical cross entropy score. We will not go into detail about how these software work yet we will touch on what the categorical cross entropy error is.

Categorical cross entropy or CCE is calculated by

 CCE = CCE(y, yˆ) = − (y1 log(ˆy1) + y2 log(ˆy2) + . . . + ys log(ˆys)).

In layman’s terms it is the difference between the prediction and the actual category (y and $\hat{y}$ ) it can be thought of the error of the model predicting the label. The mean categorical cross entropy error is the average cross entropy errors for all the examples in the set

Some important properties of CCE to note are:

CCE >= 0

if $\hat{y}$ is equal to y then CCE(y, $\hat{y}$ ) = 0

Results

Figure 6 below displays a graph of our dataset with each dot being a news article in our dataset. And the vectors of each article on the two axises.



Figure 6

As seen on the graph, news articles of the same category are grouped together as they have very similar tf-idf feature vectors. Using logistic regression decision boundaries are created around the data points, this is what is used to predict the category of the articles with the articles following in specific boundaries being predicted to be the corresponding category. As mentioned previously we used training and validation sets. On the training set we had a mean CCE score of 97+%, and the accuracy of the model on the validation set can be seen in the confusion matrix below.



The confusion matrix shows the number of data points that were correctly and incorrectly predicted. As seen above there were only 6 articles that were incorrectly categorized in the validation set. Furthermore, sport and entertainment categories both had 100% accuracy.

Conclusion

Machine learning is a vast field with many world changing applications. In this article we explored the subset of Natural language processing and its efficacy to categorize news articles.

References

[1] Andriy Burkov. The hundred-page machine learning book, volume 1. Andriy Burkov Canada,
2019.
[2] Sebastian Raschka. Python machine learning. Packt publishing ltd, 2015.

[3] Cha Zhang and Yunqian Ma. Ensemble machine learning: methods and applications. Springer, 2012.