Documents Topic Categorization

Khoula K. Al-Kharusi
Computer Science, Sultan Qaboos University

Abstract

Document categorization is the process of assigning categories or classes to documents depending on their subject matter to make them easier to manage, search, filter, or analyze. Commonly, document categorization tasks are divided into text and visual classifications in therm of data. This research problem concentrated on categorizing documents relaying in the text only. Two supervised baseline models are used to classify news documents into their ten categories. The documents were vectorized using TF-IDf and the two classifier models are logistic regression and support vector machines. The models were tuned using grid search method and evaluated by of accuracy, sensitivity, specificity, F-score and AUC.

Keywords: document categorization; classification; SVM; logistic regression

1 Introduction

Text classification is a subset of a broader problem of automatic content analysis (1). Automated content methods can make it possible to analyse large-scale text collections without massive funding support. Text classification is one of the fundamental tasks in Natural Language Processing (NLP). The goal is to organise texts into categories. It has wide applications, including topic labelling, sentiment classification, and spam detection(5). For a document to be classified under a given rubric, it must be genuine that its subject matter relates to that category assigned. This is a relatively easy decision for a human being to make in most cases. The question being raised is whether a model can learn to determine the subject content of a document and the category into which it should be classified. This question has practical implications. Vast quantities of literature are inundating technical libraries. Automatic procedures must be developed to handle the initial processing of this material. Unless such help is provided, there will be an unreasonable delay in processing the document prior to its distribution to the interested user(2). In this paper, baseline classification models are used to catigorise the text documents into ten different subjects.

1.1 Problem Statement

Document categorization is the process of assigning categories or classes to documents depending on their subject matter to make them easier to manage, search, filter, or analyze. Commonly, document categorization tasks are divided into text and visual classifications in therm of data. Supervised methods for document categorization are referred to as document classification, and it requires prior knowledge of documents classes to train the supervised model. While the unsupervised categorization is called clustering, in which the model learns to group similar documents. This research problem concentrated on categorizing documents relaying in the text only. Two supervised baseline models are used to classify news documents into their ten categories. The labels are non-overlapping, so each document is labelled by one subject category.

1.2 Objectives

This research aims to build logistic regression and support vector machine models that accurately classify news documents to their subject category. The two models have to be tuned to find the best hyper-parameters.

1.3 Motivation

Technical libraries are flooded with a massive amount of literature. Automated techniques must be implemented to handle processing this data. There will be an excessive delay in processing the document manually. Therefore, This project is a step toward the automation of categorizing text documents.

2 Method

2.1 Document vectorizaion

The methods of representing the text to numeric data are called text representation or vectorising. There are many approaches for document representation, starting with the most basic and modest methods like Bag-of-Words moving towards more complex approaches with neural networks. In this project a method called TF-IDF is used.

2.1.1 TF-IDF

TF-IDF stands for term frequency-inverse document frequency. It is a matrix where every entry represent the TF-IDF of the term and it is the result of multiplying the term frequency by the inverse document frequency. Term frequency founded by equation1.

$$TF(w) = \frac{Number\ of\ times\ term\ w\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \tag{1}$$

Iverse document frequency is calculated by

$$idf(w) = log(N/df_i) \tag{2}$$

Where $\boldsymbol{N}$ is the total number of documents and $\boldsymbol{df_i}$ is the number of documents where a term $\boldsymbol{w}$ appeared.
TF-IDF gives a higher score for the words that appears more frequently in the document however, it decreases the score of words that appears in many documents. For example, "The" is more likely to be more frequent in all documents, thus, it gets a lower score in the TF-IDF matrix.
TF-IDF equation:

$$W_{ij} = tf_{ij} \times log(N/df_i) \tag{3}$$

Where $\boldsymbol{i}$ is the $\boldsymbol{i}^{\ th}$ term and $\boldsymbol{j}$ is for the $\boldsymbol{j}^{\ th}$ document.

2.2 Classification Models

2.2.1 Logistic Regression

Logistic regression is a statistical classification model of a generalized linear regression (4). The linear combination of the inputs is computed as

$$z = w^T X \tag{4}$$

The to find the probability the linear function z pass this through a sigmoid function that ensures *sigm(z)* ϵ *[0, 1]*.

$$sigm(z) = \frac{1}{1 + e^{-z}} \tag{5}$$

The probability is given by

$$P(y|X) = Ber(y|sigm(w^T x) \tag{6}$$

Where *Ber* is Bernoulli distribution. The loss function of the logistic regression is cross entropy and from there the cost function of the models is

$$L(y, \hat{y}) = -[y\ log(\hat{y} + (1 - y)\ log(1 - \hat{y}))] \tag{7}$$

Adding a regularization term that is controlled by regularization parameter λ.

$$J(w) = L(y, \hat{y}) + \lambda\ penalty \tag{8}$$

2.2.2 Support Vector Machine (SVM)

SVM is one of the most significant developments in pattern recognition. The simplest form of SVM is a linear discriminant function that separates classes. For generalization, the hyperplane is selected to be as far as possible from any sample. The distance between the datapoints and the hyperplane is called the margin.

$$margin = \frac{2}{||w||} \tag{9}$$

w is the coefficients of the hyperplane. Convert it to a minimization problem to deal with is as a loss function.

$$J(w) = \frac{1}{2}||w||^2 \tag{10}$$

$$constrain\ \ z_i(w^T x_i + w_0) \geq 1 \quad \forall i \tag{11}$$

J(w) is quadratic function with single global minimum. using Kuhn-Tucker theorm (KKT) to convert the problem function to the form of

$$maximize\ L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j z_i z_j x_i^T x_j \tag{12}$$

Where α is a value assign to each sample, $\alpha = 0$ for all samples accept the support vector samples (The closest points to the margin). This is the reason why adding data out of margin will not affect separating line. For soft margin a slack variable is added for each sample. this will change the constrain to be:

$$constrain\ \ z_i(w^T x_i + w_0) \geq 1 - \xi_i \quad \forall i \tag{13}$$

and the cost function:

$$J(w) = \frac{1}{2}||w||^2 + \beta \sum I(\xi > 0) \tag{14}$$

$$I(\xi > 0) = 1\ \ if\ \ \xi > 0\ \ I(\xi > 0) = 0\ \ otherwise \tag{15}$$

SVM avoid curse of dimensionality from lilting data into a higher dimensional space by enforcing largest margin and the computation in the higher dimensional case is performed only implicitly through the use of kernel functions. Kernel is a similarity measure (modified dot product) between data points.

$$maximize\ L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j z_i z_j K(x_i, x_j) \tag{16}$$

The aim is to find Kernel function that $\mathrm{k(x_i,\ x_j)} = \varphi(\mathrm{x_i})^T\ \varphi(\mathrm{x_j})$

commonly used kernels:

Linear

$$k(x_i, x_j) = x_i.x_j \tag{17}$$

Polynomial of power p

$$k(x_i, x\_j) = (1 + x_i.x_j)^P \tag{18}$$

Radial-basis function (rbf)

$$k(x_i, x_j) = exp[-\gamma.\ ||x_i - x_j||] \tag{19}$$

Sigmoid

$$k(x_i, x_j) = tanh(\beta_0\ x_i\ x_j\ + \beta_1) \tag{20}$$

3 Experiments

3.1 Dataset

The dataset was taken from Kaggle website and can be found here. It contains ten folders, each of a different news group. One hundred text files of news documents are in each folder. The dataset can be described in the table1:

Category	No. Documents	Document Format
Space	100	Text file
Politics	100	Text file
Sport	100	Text file
Technologies	100	Text file
Historical	100	Text file
Medical	100	Text file
Graphics	100	Text file
Entertainment	100	Text file
Food	100	Text file
Business	100	Text file

Table 1: Row dataset description

3.1.1 Data preprocessing

The text was preprocessed before it has been vectorized to shrink the size of the vectors and enhance the results:

1. **Lowercase conversion:** therefore the same word appears ones only
2. **Tokenization:** each word as a token with removing punctuation marks
3. **Removing stop words:** stop words frequently appears throughout the text, although they do not add any meaning out of their context
4. **Stemming and Lemmatization:** representing a set of words by their common root

After this preprocessing, the text was vectorized using TF-IDF limited with the most 500 frequent tokens in the corpus. Finally, the dataset was structured and saved in (1000*501) DataFrame containing five hundred features (document-representation of TF-IDF 3) and the numeric category label, the 501 columns. One thousand samples, each represented by a row in the DataFrame.

3.2 Experimental Setup

Python 3 has been used as a programming language for this project to process data and implement classification models because of its helpful libraries. Before implementing the models, the dataset was split to train and test sets. Afterwards, grid-search cross-validation was used to find the optimal LR and SVM models parameters. The final step was evaluating the model. In the following subsections, the details of the setup are discussed.

3.2.1 Training and Testing Sets

The dataset was split into a train set and a test set. The training set contains 80% of the whole dataset, and the testing set is the remaining 20% of the dataset. However, the data was shuffled first, and then a random 20 samples were taken from each category to be in the test set. This step was done to ensure we have an equally likely distribution of the test-train portions for each class.

3.2.2 Models Tuning

Model tuning is the process of finding the optimal hyperparameters for the model. Grid-search cross-validation has been used to tune LR and SVM classification models. Grid-search is a strategy used to search different values for model hyperparameters and choose a subset that results in a model that achieves the best performance on a given dataset. The researcher sets the values of the hyperparameters in advance for the algorithm. Then, the grid-search algorithm combines all possible sets of each hyperparameter values and evaluates the model using cross-validation. In Table 2 the hyper parameter used for grid-search to tune the classification models.

Model	Hyperparameter	Values
Logistic Regression	C	0.1, 1, 10, 100, 1000
	penalty	none, l1, l2, elasticnet
	solver	newton-cg, lbfgs, liblinear, sag, saga
Support Vector Machines	C	0.1, 1, 10, 100, 1000
	gamma	scale, 1, 0.1, 0.01, 0.001, 0.0001
	kernel	linear, rbf, poly, sigmoid

Table 2: Hyperparameter values for LR and SVM model tuning

Where the parameters are:

1. **C** is the regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.
2. **penalty** Specify the norm of the penalty for regularization.
3. **solver** Algorithm to use in the optimization problem.
4. **gammea** Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.
5. **kernel** Specifies the kernel type to be used in the algorithm.

3.2.3 Classification Performance Evaluation Metrics

There are a number of evaluation metrics for classification. The confusion matrix 1 is one way of summarizing the classification results of a model. The rows represent the predicted label while the columns represent the actual label. It has 4 entries: TP and TN donate the number of positive and negative correctly classified instances while FP and FN donate the number of positive and negative miss-classified instances (3).

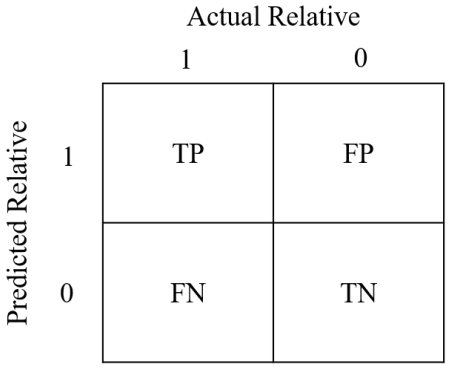

Figure 1: Confusion Matrix

Accuracy measures the ratio of correct predictions over the total number of instances evaluated.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{21}$$

Other metrics are preferred for in-balanced datasets, we list them in the following sections:

Precision measures the positive instances that are correctly predicted from the total positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

Recall measures the fraction of positive data points that are correctly classified.

$$Recall = \frac{TP}{TP + TN} \tag{23}$$

False Positive Rate FPR measures the fraction of negative instances incorrectly identified as positive instances in the data

$$FPR = \frac{FP}{FP + TN} \tag{24}$$

Specificity or True Negative Rate is measures the rate of correctly reject instances

$$TNR = \frac{TN}{FP + TN} = 1 - FPR \tag{25}$$

F1 Score is the harmonic mean of both Precision and Recall.

$$F1Score = 2 * \frac{precision * recall}{precision + recall} \tag{26}$$

Since our classification model categories documents to multi-classes we are going to calculate the measurements for class separately.

Area Under the Receiver Operating Characteristic Curve AUC-ROC ROC curves are two-dimensional graphs that visually illustrate a classification model's performance and trade-off. ROC graphs are constructed by plotting the true positive rate 23 against the false positive rate 24. ROC curves can be directly computed for any classification model that attaches a probability to each prediction. The ROC algorithm uses the instance probability to sweep through different decision thresholds from the maximum to the minimum ranking value in predetermined increments. The default decision threshold for most classifiers is set to 0.5. The ROC graph traces a curve from left to right (maximum ranking to minimum ranking). That means that the left part of the curve represents the model's behaviour under high decision thresholds (conservative) and the right part of the curve represents the model's behaviour under lower decision thresholds (liberal).

4 Results and Discussion

4.1 Models Tuning Results

The models' hyperparameters values were sat depending on the grid search results. The logistic regression model parameters are:

Hyperparameter	Values
C	10
penalty	l2
solver	newton-cg

Table 3: Logistic regression model's parameter

The penalty for regularization is l2-norm which is also known by Euclidean norm or Ridge norm. The solver is newton-cg which fits the multi-class categorization problem. The SVM model parameters are:

Hyperparameter	Values
C	1
gamma	scale
kernel	linear

Table 4: Logistic regression model's parameter

The inverse regularization parameter is 1 which is smaller than the LR model. The kernel is linear and it is less complected to compute compare to the other kernels.

4.2 Classification Results

To evaluate the models in the document categorization task the confusion matrix has been found for each model and it can be used to get the accuracy, sensitivity, specificity and F-score.

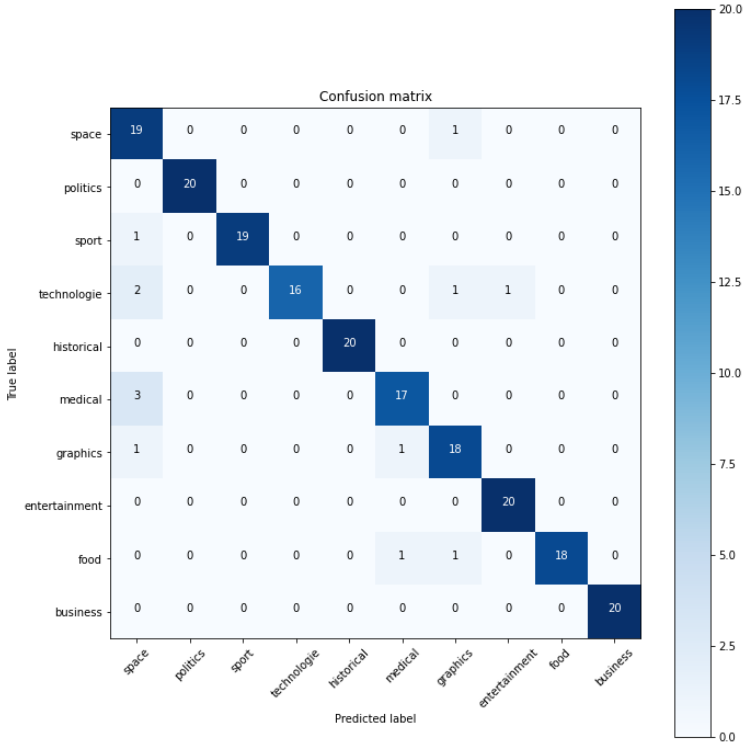

Figure 2: Logistic regression classifier confusion matrix

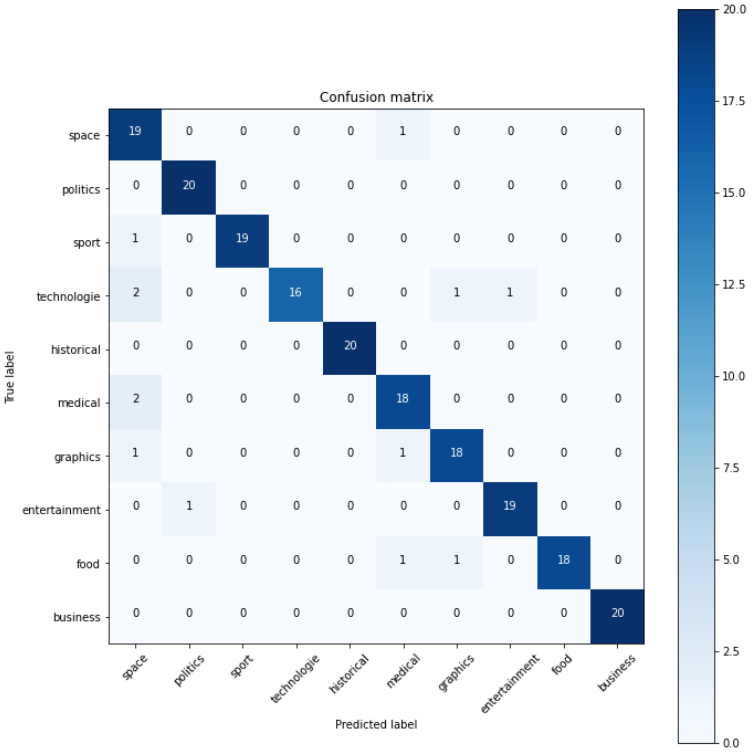

Figure 3: SVM classifier confusion matrix

The first look at the two confusion matrices; they are similar. There are many more exciting things we can get from them. Table 5 summarize the performance of the two models classification of the test set.

	LR			**SVM**		
Category	**Sensitivity**	**Specificity**	**F-score**	**Sensitivity**	**Specificity**	**F-score**
space	0.95	0.994	0.826	0.95	0.994	0.844
politics	**1.00**	**1.00**	**1.00**	1.00	1.00	0.976
sports	0.95	0.994	0.974	0.95	0.994	0.974
technology	0.80	0.994	0.889	0.80	0.994	0.889
historical	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
medical	0.85	0.994	0.872	0.90	0.994	0.878
graphics	0.90	0.994	0.878	0.90	0.994	0.90
entertainment	1.00	1.00	0.976	0.95	0.994	0.95
food	0.90	0.994	0.947	0.90	0.994	0.947
business	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Average	0.935	0.996	0.936	0.935	0.996	0.936

Table 5: LR and SVM performance evaluation in test set classification

From the table above and the confusion matrices, both models categorize historical and business newsgroups perfectly. The models were able to find all the two categories of documents and never assign other documents wrongly to those two groups. This is also the case for the political class in the logistic regression. The total accuracy of LR model and SVM model is 93.5% . The misclassification distributions in the confusion matrices are akin. The document's content may explain this. Two documents out of the 200 test samples are classified differently by each model, one from the entertainment newsgroup and the other from the medical newsgroup.

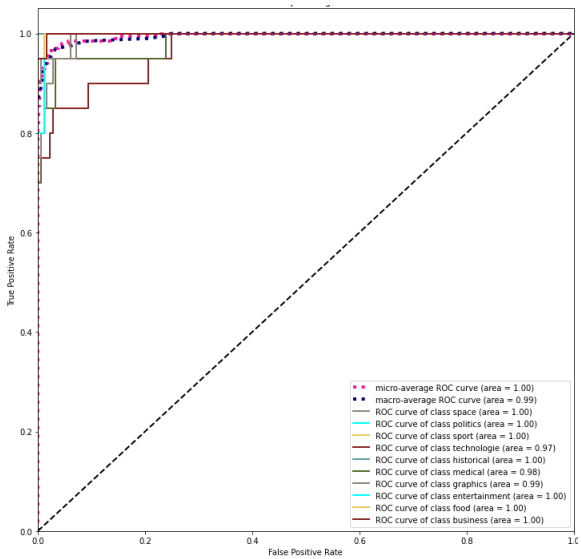

Figure 4: LR model ROC curve

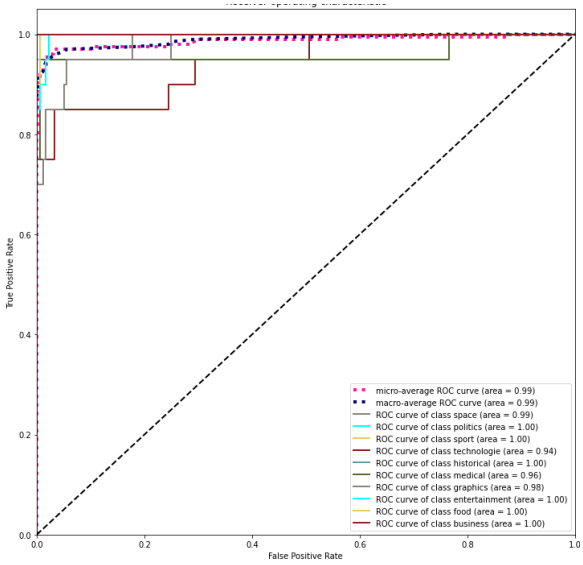

Figure 5: SVM model ROC curve

The two models ROC curve are almost perfect for most classes. The area under the LR curve is 0.9938 while the area under the SVM model ROC curve is 0.987. The LR model is slightly better in term of ROC.

5 Conclusion

This project aimed to build two classification models that accurately classify news documents to their subject category and compare their performance. The two models of logistic regression and support vector machine showed excellent results with text classification where both models accuracy reached 93.5% even though they are considered baseline models. There was no significant diversity between the two models. However, the area under the LR model ROC curve is slightly greater by 0.0068.

References

H. Borko and M. Bernick. Automatic document classification. *J. ACM*, 10(2):151–162, apr 1963.

J. Grimmer and B. M. Stewart. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3):267–297, 2013.

M. Hossin and M. Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015.

K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.