**Spam Comment Classification**

By Tegar Dani Pratama

**Objective**

Make a classifier to classify news title. There are 4 categories namely Business, Entertainment, Medical, and Technology.

**Flow Chart**

* Precision
* Recall
* F1 Score

Data

Data Cleaning

Cleaning

Evaluation

Choosing Model

* Logistic Regression
* Naive Bayes
* Random Forest Tree
* Remove Non-title text
* Remove duplicates
* Normalize data:
* Lowercase all words
* Convert shortform into longform
* Remove punctuation and symbols

**Data Cleaning**

From the comment data. It could be seen that there are still many duplicates (434 of them). Duplicates should be removed.

There are also strange News title such as:

* ‘article requested cannot be found’,
* ‘Posted by ....’,
* ‘PR Newswire’,
* ‘Business Wire’,
* ‘click the phrases to see a list’,
* ‘Share this on:’.

There should be many more but i have no knowledge of them so i just removed the above list from the data sets

After that, the dataset should be normalized. Normalizing datasets includes: transform the text into lowercase, remove symbols, remove punctuations, convert shortform into long form, and remove stopwords.

**Model**

Preporcessing method:

1. **Count Vectorizer**

Tokenize the collection of documents and form a vocabulary with it and use this vocabulary to encode new documents. We can use CountVectorizer of the scikit-learn library. It by default remove punctuation and lower the documents.

1. **TF-IDF** (Term Frequency times Inverse Document Frequency)

a metric that represents how 'important' a word is to a document in the document set

In choosing model I decided to use feature-based model. The simplest and traditional models of machine learning.

We train Logistic Regression, Naive Bayes, Random Forest Tree on the dataset.

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F-1 Score** |
| **Logistic Regression** | **0.92** | **0.89** | **0.90** |
| Multinomial Naive Bayes | 0.91 | 0.87 | 0.89 |
| Random Forest Tree | 0.88 | 0.85 | 0.86 |

**Conclusions**

On average our models are about 90.75% accurate. While this may mean that the machine cannot predict every comments with 100% accuracy. It can really depend on the data and how it has been processed.

Out of the 4 used in this project, the most accurate and precise was the Logistic Regression, with accuracy of 92%.

Logistic Regression highest recall, with recall score 89%. This means that the models actually calculates how many of the Actual Positives our model capture through labeling it as Positive (True Positive).

Logistic Regression has the highest F1 score, with f1 score 90%, which means that it defines a relationship between Recall and Precision of a particular model. F1 Scores might be a better measure to use if we need to seek a balance between Precision and Recall and if there is an uneven class distribution (a large number of Actual Negatives).

So the best model for this dataset is using Logistic Regression.