**Spam Comment Classification**

By Tegar Dani Pratama

**Objective**

Make a classifier to decide a comments whether it’s spam or not. In the field Class, 0 means not spam and 1 means spam.

**Flow Chart**

* Precision
* Recall
* F1 Score
* Logistic Regression
* Naive Bayes
* Random Forest Tree
* SGD
* Remove duplicates
* Replace Strange String
* Denoise data:
* Strip html tag

Data

Data Cleaning

Cleaning

Evaluation

Choosing Model

**Data Cleaning**

From the comment data. It could be seen that there are still many duplicates (100 of them) and noise. Such as typo, converted symbols, and html tags. The first step is to remove them.

Usually NLP includes normalizing data such as removing stopwords and transforn word into stems/lemma. I did them at first but after normalizing the data, But it reduced the accuracy of the model. So, i decided not to normalize them.

**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, and SGD on the dataset.

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F-1 Score** |
| Logistic Regression | 0.95 | 0.92 | 0.93 |
| Multinomial Naive Bayes | 0.93 | 0.92 | 0.93 |
| Random Forest Tree | 0.91 | 0.88 | 0.89 |
| **Stochastic Gradient Descent** | **0.95** | **0.93** | **0.94** |

**Conclusions**

On average our models are about 93.5% 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 Stochastic Gradient Descent (SGD), with accuracy of 95%.

SGD highest recall, with recall score 93%. This means that the SGD actually calculates how many of the Actual Positives our model capture through labeling it as Positive (True Positive).

SGD has the highest F1 score, with f1 score 94%, 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 SGD.