

Identifying spam messages using R and Pandas over Docker Swarm

SAGAR VORA^{1,*} AND RAHUL SINGH¹

¹ School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

* Corresponding authors: vorasagar7@gmail.com, rahul_singh919@yahoo.com

project-1, March 29, 2017

A classification model shall be built by working on a training data set of 5574 text messages, each marked as a spam or a legitimate message. The model shall be used to correctly predict the class of any new incoming text message as a spam or a legitimate one. The application shall be deployed using [1] Docker Swarm while data manipulation and classification shall be done using Pandas and R in conjunction.

© 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

Keywords: Docker, Swarm, R, Pandas

<https://github.com/cloudmesh/sp17-i524/raw/master/project/S17-IR-P007/report/report.pdf>

test

1. INTRODUCTION

To address the problem of incoming spam messages, a model shall be developed using the Bayesian Classification technique to correctly classify each incoming email/text message as a spam or a legitimate one. The model aims at developing a message filter that shall correctly classify messages based on word probabilities that are extracted from the training dataset. The training dataset to build the model consists of 5574 message records. Dataset taken from [2]. The training process shall use the cross-validation feature provided by R to build the classification model and use Bayes theorem of conditional probability to predict the class of each incoming message.

2. DESIGN

2.1. Building the Classification model

2.1.1. CrossValidation for the training data

To develop an efficient training model, we shall partition the data into 2 subsets - training data and classification data. We shall choose one of the subsets for training and other for testing. In the next iteration the roles of the subsets shall be reversed, i.e the training data becomes the classification one and vice versa. This operation shall be carried out until each individual record is used both as a classification and training record. We shall use the cross validation feature provided by R for this subsampling. This subsampling technique handles the underfitting problem and guarantees an effective classification model.

2.1.2. Training process

Content of each of the spam marked messages shall be processed through Naive Bayes Classifier. The classifier shall maintain a bag of words along with the count of each word occurring in the spam messages. This word count shall be used to calculate and store the word probability in a table that shall be cross-referenced to determine the class of the record on classification data [3].

A selected few words have more probability of occurring in a spam messages than in the legitimate ones. Eg: The word "Lottery" shall be encountered more often in a spam message. The classifier shall correlate the bag of words with spam and non-spam messages and then use Bayes Theorem to calculate a probability score that shall indicate whether a message is a spam or not. The results shall be verified with the results available on the training dataset and the classifier accuracy shall be calculated. The classifier shall use the Bayesian theorem over the training dataset to calculate probabilities of such words that occur more often in spam messages and later use a summation of scores of the occurrence of these word probabilities to estimate whether a message shall be classified as spam or not. After working on several samples of the training dataset, the classifier shall have learned a high probability for spam based words whereas, words in legitimate message like family member or friends names shall have a very low probability of occurrence.

2.2. Classifying new data

Once the training process has been completed, the posterior probability for all the words in the new input email is computed using Bayes theorem. A threshold value shall be defined to classify a message into either class. A message's spam probability is computed over all words in its body and if the sum total of the

probabilities exceeds the predefined threshold, the filter shall mark the message as a spam [4].

A higher filtering accuracy shall be achieved through filtering by looking at the message header i.e the sender's number/name. Thereby if a message from a particular sender is repeatedly marked as spam by the user, the classifier need not evaluate the message body if it is from the same sender.

3. DISCUSSION

TBD

4. DEPLOYMENT

Our application will be deployed using Ansible [5] playbook. Automated deployment should happen on two or more nodes clouds or on multiple clusters of a single cloud. Deployment script should install all necessary software along with the project code to the cluster nodes.

5. CONCLUSION

TBD

6. ACKNOWLEDGEMENT

We acknowledge our professor Gregor von Laszewski and all associate instructors for helping us and guiding us throughout this project.

7. APPENDICES

TBD

REFERENCES

- [1] "Docker swarm," Web Page, accessed: 2017-03-10. [Online]. Available: <https://docs.docker.com/engine/swarm/>
- [2] "SMS spam collection dataset," Web Page, accessed: 2017-03-10. [Online]. Available: <https://www.kaggle.com/uciml/sms-spam-collection-dataset>
- [3] J. Provost, "Naive-Bayes vs. Rule-Learning in Classification of Email," in *Artificial Intelligence Lab*. The University Of Texas at Austin: The University Of Texas, 1999, accessed: 2017-03-10. [Online]. Available: <http://mathcs.wilkes.edu/~kapolka/cs340/provost-ai-tr-99-281.pdf>
- [4] Wikipedia, "Naive bayes spam filtering," Web Page, January 2017, accessed: 2017-03-10. [Online]. Available: https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering
- [5] "Ansible, deploy apps. manage systems. crush complexity," Web Page, accessed: 2017-03-12. [Online]. Available: <https://www.ansible.com/>