Deploying a spam message detection application using R over Docker and Kubernetes

SAGAR VORA[1,*] AND RAHUL SINGH[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: vorasagar7@gmail.com, rahul_singh919@yahoo.com

project-P007, April 30, 2017

In the last few decades, online spam has become one of the major problem for the sustainability of the Internet. Due to the excessive amount of spams, the quality of information available on the Internet has reduce drastically. Moreover spam messages are also creating problems among the various search engines available and the web users. This report aims at developing an application which would detect spam messages from actual meaningful messages using Pandas and R. For the purpose for parallelizing the process, we would deploy the application using Docker containers on the Kubernetes cluster using ansible scripts which would automate the deployment.

Keywords: Docker, Ansible, Kubernetes, R, Pandas, Spam, <add spam detection algorithms>

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

1. INTRODUCTION

Today, the Internet [1] has been adopted rapidly in the day to day life of people. It has provided a platform for information generation and consumption. Moreover, it is used on a daily basis to search for information and acquire knowledge. The online encyclopedia Wikipedia™ [2] provides a good example of a more socialized Internet because the content within Wikipedia™ is collectively generated by its users, rather than webmasters or designated editors. The ease with which content can be generated and published has also made it easier to create spam. Spam can be stated as any information which does not add value to a user of the web. Messages which are inappropriate, unsolicited, repeated and irrelevant can be all classified as spam.

So in this report, we are provinding an application that would identify valid messages and spam messages from a given dataset. For spam detection, we are using various techniques like Bayesian, <text here> ..... Moreover, deploying our application using Docker [3] containers on the Kubernetes [4] cluster will give a distributed approach.This would also speed up the process of identifying the spam messages. We have also deployed it on different cloud environments like Chameleon, JetStream, FutureSystem and have performed benchmark analysis of the application. This would let us the time taken by the algorithm on these cloud solutions.

Name of the Technology	Purpose in the Project
R	data analytics
Docker	container for the applicat
Kubernetes	cluster creation and manag
Cloudmesh Client	An client application used to ssh in various
Ansible	Automation language to deploy application

Fig. 1. Technologies used in the Project

2. SOFTWARE STACK

3. WHY KUBERNETES

<write why we choose Kubernetes>

4. WHAT IS KUBERNETES

<what is Kubernetes>

5. ARCHITECTURE

<Architecture of Kubernetes>

6. ANSIBLE

No one likes repetitive tasks, so with Ansible [5], IT admins can begin automating away the drudgery from their daily routine tasks. Ansible is a simple automation language that can perfectly describe an IT application infrastructure. Ansible is an open source automation engine which can be used to automate cloud provisioning, configuration management, and application deployment. It can also perform more advanced IT tasks such as continuous deployment or rolling out updates with zero downtime.

A major difference in Ansible and many other tools in the space is its architecture.

6.1. Architecture

Ansible is an agentless tool,it doesn't requires any software to be installed on the remote machines to make them manageable. By default is manages remote machines over SSH or WinRM, which are natively present on those platforms [**?**].

Like the other configuration management software, Ansible distinguishes between two types of servers: one being the controlling machines and other being the nodes. Ansible uses a single controlling machine where the orchestration begins. Nodes are then controlled by a controlling machine over SSH [**?**]. The location of the nodes are described by the inventory of the controlling machine.

Ansible modules are deployed by Ansible over SSH. These modules are temporarily stored in the nodes and communicate with the controlling machine through a JSON protocol over the standard output

6.2. Playbooks

Playbooks [6] are Ansible's configuration, deployment, and orchestration language. They let us control the remote systems with a policy which we might want them to enforce. If Ansible modules act as tools in your workshop, then playbooks are your instruction manuals, and your inventory of hosts are your raw material. Playbooks can be used to manage configurations of and deployments to remote machines. They can sequence multi-tier rollouts involving rolling updates, and can delegate actions to other hosts, interacting with monitoring servers and load balancers.

6.3. Ansible Galaxy

6.4. Pandas

Pandas is an opensource Python library that provides data analysis functionality with Python. Python initially lacked data analysis and modeling capability. Pandas filled out this gap by providing essential analytic functions thus saving the need to switch to a more domain specific language for data analysis.

6.5. R

R is a language and environment for statistical computing and graphics [7]. Pandas does not provide a significant statistical modeling environment as it is still a work in progress. R provides a variety of statistical model analysis, classification, clustering and graphical techniques to provide this environment. Integrating Python's efficiency with R's capability allows us to build a highly a desirable analysis model for our application.

6.6. Docker

Docker allows application developers to package their applications into isolated containers. A container comprises of only the libraries and settings that are required to make the software work. Docker automates the repetitive tasks of setting up and configuring development environments thus allowing developers to focus only on building software. A dockerized application can simply ship between platforms as the complexity of software depencies is handled by the container.

6.7. Kubernetes

Kubernetes is an open-source platform which helps in automating deployment, scaling, and operations of application containers across clusters of hosts. Kubernetes helps in faster deployment of application and scaling them on the fly. Moreover it optimizes the use of hardware by using the resources which are needed. A Kubernetes cluster can be deployed on either physical or virtual machines. We will be using Minikube which is a lightweight Kubernetes implementation which creates a VM on the local machine and deploys a simple cluster containing only one node. The Minikube CLI provides basic bootstrapping operations for working with the cluster, including start, stop, status, and delete commands.

7. DESIGN

7.1. Building the Classification model

7.1.1. CrossValidation for the training data

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 [8]. 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. < I just copy pasted the above paragrah here> Donno what will come here.

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.

7.1.2. 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 occuring 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 [9].

A selected few words have more probability of occuring 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 occurence 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 occurence.

7.2. 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 [10].

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.

8. DISCUSSION

TBD

9. 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 Kubernetes cluster nodes using the Docker image.

10. CONCLUSION

TBD

11. ACKNOWLEDGEMENT

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

12. APPENDICES

TBD

REFERENCES

[1] "What is internet?" Web Page, accessed: 2017-04-12. [Online]. Available: http://searchwindevelopment.techtarget.com/definition/Internet

[2] "Wikipedia," Web Page, accessed: 2017-04-12. [Online]. Available: https://www.wikipedia.org/

[3] "What is docker?" Web Page, accessed: 2017-04-02. [Online]. Available: https://www.docker.com/what-docker

[4] "Kubernetes," Web Page, accessed: 2017-03-10. [Online]. Available: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

[5] "Ansible, deploy apps. manage systems. crush complexity," Web Page, accessed: 2017-04-12. [Online]. Available: https://www.ansible.com/it-automation

[6] "Playbook," Web Page, accessed: 2017-04-12. [Online]. Available: http://docs.ansible.com/ansible/playbooks.html

[7] "R:what is R?" Web Page, accessed: 2017-04-02. [Online]. Available: https://www.r-project.org/about.html

[8] "SMS spam collection dataset," Web Page, accessed: 2017-03-10. [Online]. Available: https://www.kaggle.com/uciml/sms-spam-collection-dataset

[9] 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

[10] Wikipedia, "Naive bayes spam filtering," Web Page, January 2017, accessed: 2017-03-10. [Online]. Available: https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering