

# Cloud based Naïve Bayes statistical email filtering on Enron Dataset using AWS EMR and EC2

Adarsh Baluni, Ishani sheth, Pujal Trivedi, Tanushri Khandekar  
University of Southern California  
90007 Los Angeles, CA, USA

**Abstract**—In this report, we perform email classification. In today's global era, email is one of the fastest form of communications. Due to the increase in email users these days, there are more spam emails. Machine learning techniques can analyze the data and create a model to predict unseen emails as spam or not spam. We deploy a GUI interface and a Naïve-based Machine Learning technique to evaluate and classify emails. Both the GUI and Naïve based techniques are deployed using Amazon Web Services. Also the cloud performance is evaluated on different cluster sizes.

## 1. PROBLEM STATEMENT

Nowadays email is one the popular communication mechanism as Internet becomes globally available. As the number of emails increases, the management of the emails becomes an important issue. The easy access of emails can give freedom to misuse it. Among the misuse performed, email spamming is the biggest misuse or nuisance to receiver. Spam can be defined as junk mail or unsolicited bulk email which involves almost identical messages sent to multiple people via emails. Spam mails also contain disguised links which appear to be similar to familiar and commonly used websites but that leads to phishing websites or malware sites.

Emails can be viewed as text classification as most of the emails contain text. Email classification is challenging due to the volume and variety of features in the dataset. Also, we have large number of documents which require classification. Large number of features make document classification limited. In majority of the document datasets, a small portion of the total features might be useful in labeling the document, it is also possible that using all the features may affect performance adversely. The volume and quality of training dataset is crucial in deciding the performance of feature selection and text classification algorithms. A good training dataset for each category includes all the important terms/features and their probable distribution in the category. [1]

## 2. METHODS APPLIED

### 2.1 NAÏVE BAYESIAN(NB)

In order to classify emails, some classification techniques like Naïve Bayesian (NB), Neural Network (NN) and Support Vector Machine(SVM) are most commonly used. In this report, we have used the Naïve Bayesian (NB) technique to classify email.

Naïve Bayes is a machine learning technique which is based on Bayes' theorem. Naive Bayes Algorithm gives a way of calculating the posterior probability,  $P(c|x)$ , from  $P(c)$ ,  $P(x)$ , and  $P(x|c)$  where,

- $P(c|x)$  is the posterior probability of class given predictor.
- $P(c)$  is the prior probability of class.

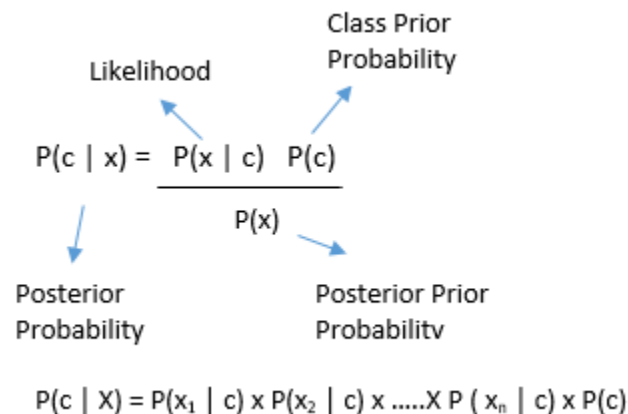
- $P(x|c)$  is the likelihood which gives predictor given class's probability..
- $P(x)$  is the prior probability of predictor.

Naïve Bayes classifier gives assumption that the effect of the predictor(x) on a given class(c) is independent of the values of the other predictors. This is called class conditional independence.

Conditional Independence Definition: If we have random variables like X,Y and Z , we say X is conditionally independent of Y given Z, if X is independent of value of Y and Z; that is

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Naïve Bayes Formulation:



## 2.2 PYTHON

Python is a high level programming language which supports dynamic name resolution i.e. it binds the method and variable names during the execution of the program. This feature of python makes it more popular among the users nowadays.

We have implemented the controllers for our project using the python programming language. The controller creates a model file using a labeled training dataset which in turn classifies the incoming email as a spam or ham.

## 2.3 PHP

PHP stands for Hypertext Preprocessor. PHP is a scripting language which is widely used for web-development. It is an open source scripting language. The scripts are executed on the servers.

We have used PHP to develop a graphical user interface. The LAMP/WAMP/XAMPP server are required for implementing the PHP script based on the operating system that we are using. The PHP script mainly takes the email as an input from the user, process it and converts it to a text file which will be processed by the controller implemented in python and in turn returns the result that is returned by the controller.

## 2.4 MAPREDUCE

MapReduce refers to two separate tasks that Hadoop program performs namely: Map task and Reduce Task.

In MapReduce, the input data are first partitioned into chunks/blocks of data files. Each Map task processes a single chunk of file and the intermediate map are combined by the Combiner (optional) and stored into a temporary file on the local file system. Each reduce task in turn processes the output from each map task respectively. The reduce phase includes the process of shuffle and sort. Finally the reduce phase processes the chunk and writes the final results to HDFS.

## 3. WORKFLOW PLAN

Python scripts are used to execute the Naïve Bayes. Two python scripts one for Naïve Bayes learn and one for Naïve Bayes Classifier are deployed. The NBLearn python script takes the training file and creates a ModelFile that contains feature count and prior probabilities of spam-not spam emails. Then NBClassify takes the ModelFile and classifies the Email test file created on S3 cloud.

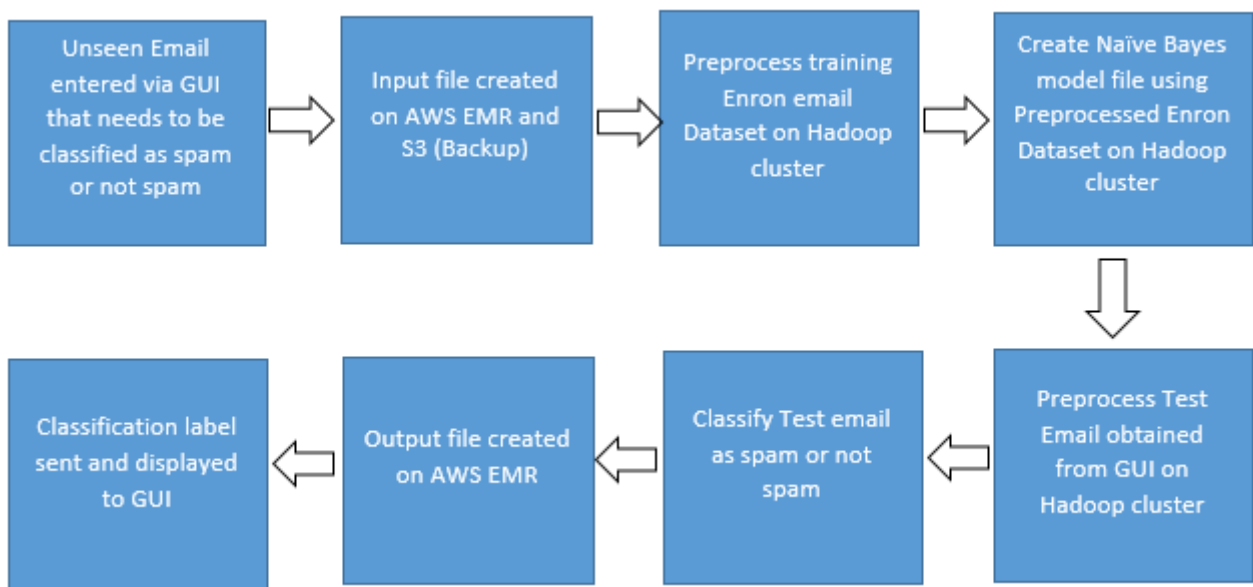


Fig 3.1 Flow Diagram

## 4. AWS EXPERIMENT DETAILS

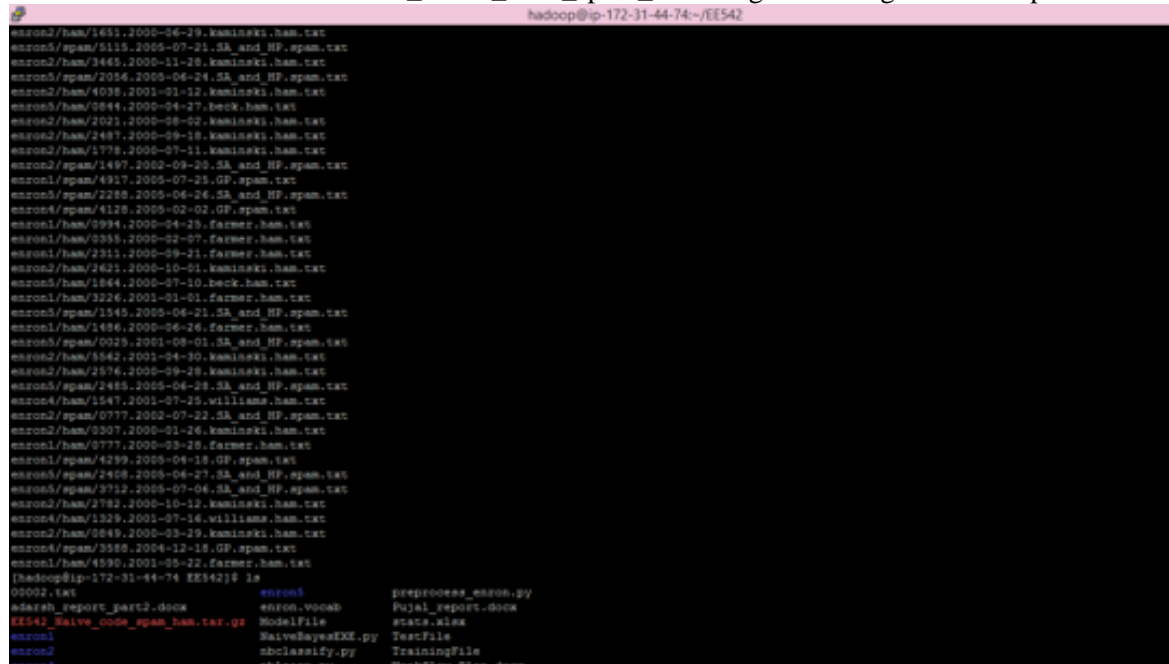
### 4.1 Running the application

- Create an EMR cluster with 1 master 1 slave. Connect to Filezilla and upload the EE542 folder to the /home/hadoop path on the Hadoop cluster that you have created. Connect to the Hadoop instance using the SSH terminal.
- Install python on the Hadoop cluster using the following command:  
`sudo yum install python34`
- Navigate to /home/hadoop/EE542 folder

- Extract the EE542\_Naive\_code\_spam\_ham.tar.gz file to get the enron1 to enron5 folders. These folders have the training files. Use the following command:

```
tar xvf EE542_Naive_code_spam_ham.tar.gz
```

The above command extracts the EE542\_Naive\_code\_spam\_ham.tar.gz file and gives the output as follows:



```

hadoop@ip-172-31-44-74:~/EE542
enron2/ham/1651.2005-06-29.kaminski.ham.txt
enron5/spam/5115.2005-07-21.SA_and_HP.spam.txt
enron2/ham/1445.2005-11-20.kaminski.ham.txt
enron5/spam/2056.2005-06-24.SA_and_HP.spam.txt
enron2/ham/4038.2001-01-12.kaminski.ham.txt
enron5/ham/0844.2005-04-27.beck.ham.txt
enron2/ham/2021.2005-08-02.kaminski.ham.txt
enron2/ham/2487.2005-09-10.kaminski.ham.txt
enron2/ham/1779.2005-07-11.kaminski.ham.txt
enron2/spam/1497.2002-09-20.SA_and_HP.spam.txt
enron1/spam/4917.2005-07-25.GP.spam.txt
enron5/spam/2288.2005-06-26.SA_and_HP.spam.txt
enron4/spam/4128.2005-02-02.GP.spam.txt
enron1/ham/0994.2005-04-25.farmer.ham.txt
enron1/ham/0355.2005-02-07.farmer.ham.txt
enron1/ham/2311.2005-09-21.farmer.ham.txt
enron2/ham/2621.2005-10-01.kaminski.ham.txt
enron5/ham/1864.2005-07-10.beck.ham.txt
enron1/ham/3226.2001-01-01.farmer.ham.txt
enron1/ham/ham
enron5/spam/1545.2005-06-21.SA_and_HP.spam.txt
enron1/ham/1886.2005-06-26.farmer.ham.txt
enron5/spam/0025.2001-08-01.SA_and_HP.spam.txt
enron2/ham/1842.2001-04-10.kaminski.ham.txt
enron2/ham/2574.2005-09-28.kaminski.ham.txt
enron5/spam/2485.2005-06-28.SA_and_HP.spam.txt
enron4/ham/1547.2001-07-25.williams.ham.txt
enron2/spam/0777.2002-07-22.SA_and_HP.spam.txt
enron2/ham/0307.2005-01-26.kaminski.ham.txt
enron1/ham/0777.2005-03-20.farmer.ham.txt
enron1/spam/4299.2005-04-18.GP.spam.txt
enron5/spam/2408.2005-06-27.SA_and_HP.spam.txt
enron5/spam/3712.2005-07-06.SA_and_HP.spam.txt
enron2/ham/2782.2005-10-12.kaminski.ham.txt
enron4/ham/1829.2001-07-16.williams.ham.txt
enron2/ham/0649.2005-09-29.kaminski.ham.txt
enron4/spam/3588.2004-12-19.GP.spam.txt
enron1/ham/4590.2001-05-22.farmer.ham.txt
[hadoop@ip-172-31-44-74 EE542]$ ls
03002.txt          enron5          preprocess_enron.py
adafsh_report_part2.docx  enron.vocab    Pujai_report.docx
EE542_Naive_code_spam_ham.tar.gz  ModelFile      stats.xlsx
enron1             NaiveBayesEXX.py  TestFile
enron2             nbclassify.py    TrainingFile
enron4             nblearn.py       Workflow_File.docx

```

Fig 4.1 Untar and unzip training dataset

## 4.2 Front End configuration:

The front end for this application is developed using PHP. To run a PHP application we need to install the LAMP server on the ec2-instance. The following steps will guide us through the same process:

1. Install the apache, PHP, MySQL server using the following command:

```
sudo yum install -y httpd24 php56 mysql55-server php56-mysqld
```

2. The Amazon Linux Apache document root is `/var/www/html`. To allow ec2-user to manipulate files in this directory, we need to modify the ownership and permissions of the directory.
3. Set the file permissions:

Add the www group to your instance: `sudo groupadd www`

Add your user (in this case, ec2-user) to the www group: `sudo usermod -a -G www ec2-user`

4. Log out and then log back in again, and verify your membership in the www group

5. `sudo chown -R ec2-user:www /var/www`
6. Using filezilla upload the team 26 folder to the `/var/www` directory
7. Open your Browser and navigate to `ec2-master-publicdns/team 26/test.php`. This page opens up as follows:



Fig 4.2 Add the email text which needs to be classified.

1. The text in the text area is uploaded as an input.txt file to the s3 bucket on clicking the submit button which is given as an input to the python controller

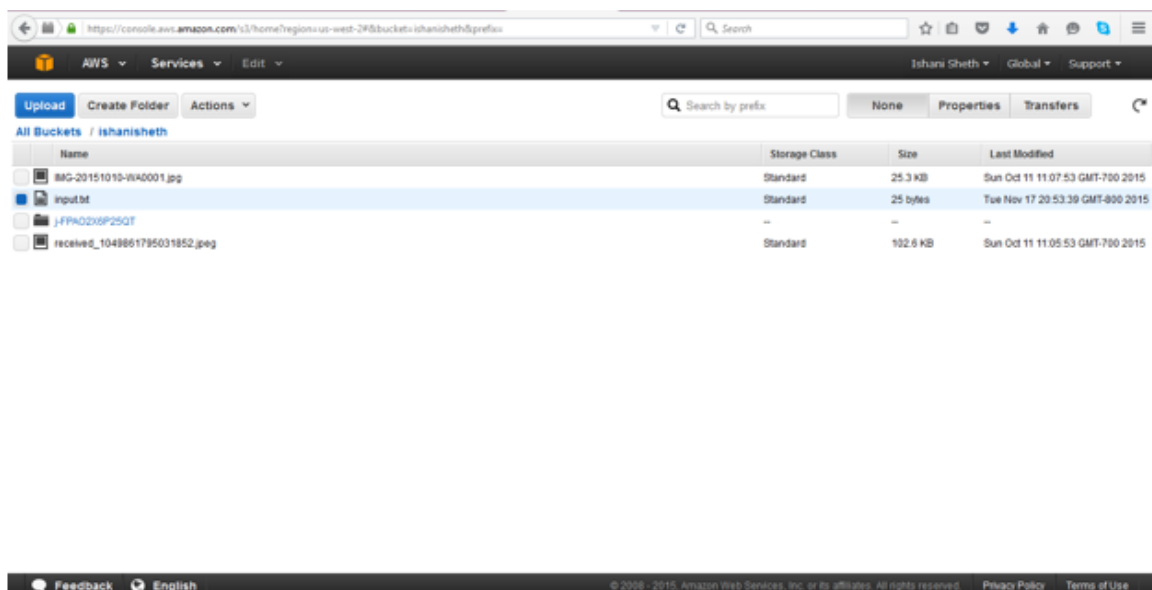
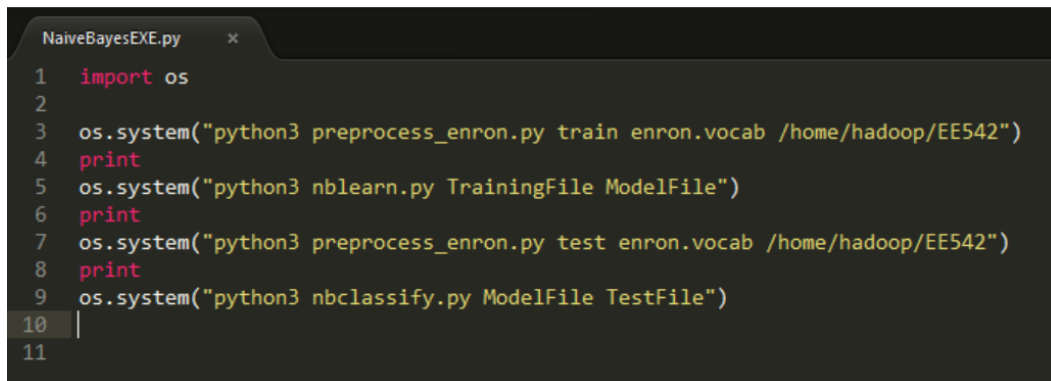


Fig 4.3 Amazon storage bucket to show input file

2. Run the following command to execute the python code which gives the output whether the input email is spam or ham:

```
python3 NaiveBayesEXE.py
```



```

NaiveBayesEXE.py
1  import os
2
3  os.system("python3 preprocess_enron.py train enron.vocab /home/hadoop/EE542")
4  print
5  os.system("python3 nblearn.py TrainingFile ModelFile")
6  print
7  os.system("python3 preprocess_enron.py test enron.vocab /home/hadoop/EE542")
8  print
9  os.system("python3 nbclassify.py ModelFile TestFile")
10 print
11 print

```

Fig 4.4 NaiveBayesEXE.py contents

Executing the above command generates the result as SPAM or HAM on clicking the Show Spam/Ham as follows:



Fig 4.5 Classified label display

## 5. AWS RESOURCES USED

### 5.1 AWS resources for Execution

**Amazon Elastic Compute Container (EC2):** Elastic MapReduce is a service available on AWS that provides for large data processing in a cost effective way. The analysis and processing of large data is done by distributing the computation tasks over a large

Features: [2]

- Intel Xeon E5-2670 v2 (Ivy Bridge) Processors or Intel Xeon E5-2670 (Sandy Bridge) Processor
- SSD storage for improved I/O performance
- Balance of CPU, memory, storage and network resources

Model	vCPU	Mem (GiB)	SSD Storage (GB)
m3.medium	1	3.75	1 x 4
m3.large	2	7.5	1 x 32
m3.xlarge	4	15	2 x 40
m3.2xlarge	8	30	2 x 80

Table 5.1 EC2 instance features [2]

**Amazon Elastic MapReduce (EMR):** Elastic MapReduce is a service available on AWS that provides for large data processing in a cost effective way. The analysis and processing of large data is done by distributing the computation tasks over a large set of virtual processing units or servers. The cluster is configured on the basis of MapReduce framework called Hadoop.

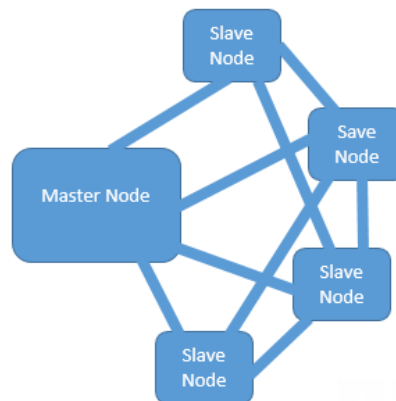


Fig 5.1 EMR Overview

Hadoop architecture uses distributed processing where tasks are divided into chunks and distributed over multiple servers for processing. This stage is called mapping. The intermediate results are then reduced to output using reducer stage. A master node is responsible for tasks distribution and among slave nodes. The following diagram shows the master-slave representation.

Amazon has made its own improvement on Hadoop for EMR. Hadoop on EMR makes use of EC2 instances as virtual servers running Linux.

## 5.2 AWS resources for Storage

**Amazon Simple Storage Service (S3):** Web service (SOAP and REST) offered by Amazon that provide online storage for file in containers called “Buckets” and enforces SSL encryption.

“S3 maintains the position of a 99.99% available and 99.999999999% durable design for storing objects throughout a period of one year. It is to be noted that for durability, there is no service-level agreement.”[4]

S3 bucket can store Files or objects upto 5 TB in size along with a metadata file of size 2KB.

## 5.3 AWS resources for Monitoring

**Amazon CloudWatch:** CloudWatch is an AWS monitoring service to check the cloud resources and applications running on AWS. It helps us in tracking, collecting and visualizing the performance matrices. It collects and monitors log files. We can select the matrices to plot graph. In this report, we will plot S3, EMR and EC2 on cloud watch along with CPU utilization and other factors.

## 6. RELEVANCE OF INDIVIDUAL BENCHMARKS

**Sorting** of our dataset that is used by our classification model. The **sort** function arranges files in order to test the performance. In many cases a sorted dataset can significantly improve performance by a big margin. For example, binary search can work best in sorted environment and performance can be improved from  $O(n)$  to  $O(\log n)$

**Aggregation** plays a key role in this project as the grouping of tokens, that determine whether an email is spam or not, is an essential step to spam-ham filtering. Aggregation is single most important feature of our model as the decision flow begins at the results of this task. [5]

The **Bayesian Classification** workload uses popular classification algorithm for data mining and knowledge discovery and is called Naïve Bayesian. In our project the importance of finding Tf-Idf (term frequency–inverse document frequency) is evident in determination of emails an incoming email as spam or not. The relevant documents are treated as training data against the incoming email and an intermediate result is generated based on this evaluation. [6]

## 7. ENRON EMAIL DATA SETS

In E-mail analysis we have used Enron Email dataset. This Dataset was originally created by Cognitive Assistant that Learns and Organizers (CALO) project. It contains all kinds of email data (professional and personal) from mostly senior management of Enron, organized into folders. Below are the features-

- No. Of users- 151 Users
- Number of input messages =~ 517,431 (.5 Million)
- Number of folders =~ 3500
- Total words in Spam: 2596038
- Total words Ham: 2817610
- Prior Probability spam: 0.50311
- Prior Probability ham: 0.49689



Since Enron has large number of users, data and Email, it was a suitable choice for Bigdata analytics and DataMining Project. The corpus is well suited for evaluation and classification of Emails.

## 8. PERFORMANCE METRICS

**Performance of the Machine learning technique** will be determined by how many emails are correctly predicted. We have actual labels in training data and using python Naïve Bayes classifier we will predict the labels.

	Actual = HAM	Actual = SPAM
Predicted = HAM	True Positive	False Positive
Predicted= SPAM	False Negative	True Negative

Table 8.1 ML performance factors

Recall-

- Definition- Ratio of correctly predicted HAM emails
- Formula-  $\text{True Positives} / (\text{True Positives} + \text{False Negatives})$

Precision-

- Definition- Ratio of correct HAM observations.
- Formula-  $\text{True Positives} / (\text{True Positives} + \text{False Positives})$ .

Accuracy-

- Definition- Ratio of correctly predicted observations
- Formula-  $\text{True Positives} + \text{True Negatives} / (\text{True Positives} + \text{False Positives} + \text{False Negatives} + \text{True Negatives})$

F1 score-

- Definition- Weighted average of Precision and Recall.
- Formula-  $2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$

**Performance of the Cloud Cluster** will be determined by how much performance was enhanced by scaling up the cluster configuration.

Time of Execution-

- Definition- Amount of time taken to execute the workload. (in seconds).

Speedup-

- Definition- Speed gain by using multiple nodes.
- Formula-  $\text{Time taken by 1 cluster} / \text{Time taken by n cluster}$

Efficiency-

- Definition- Percentage of peak performance achieved.
- Formula-  $1 / \text{Speedup}$

## 9. PLOTTED FIGURES

### 9.1 Machine Learning Graphs

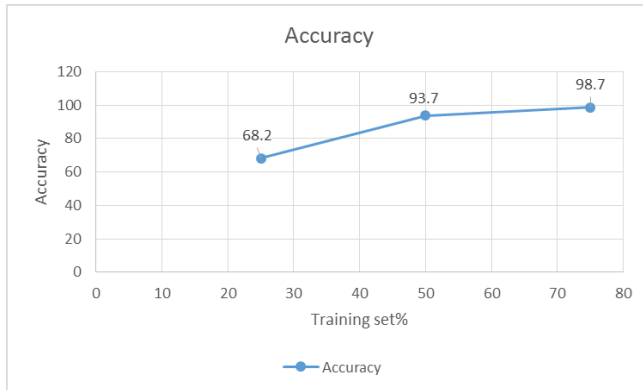


Fig 9.1 Accuracy VS Training Dataset size

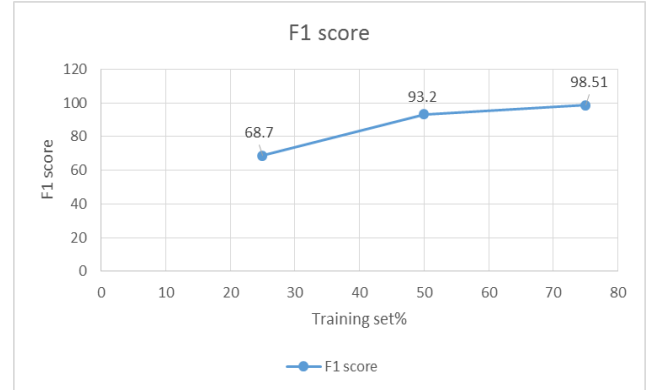


Fig 9.2 F1 score VS Training Dataset size

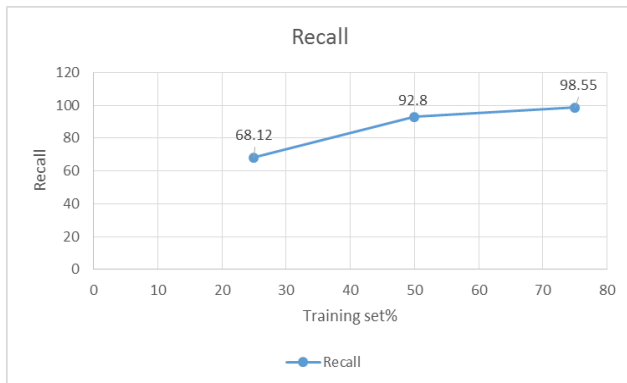


Fig 9.3 Recall VS Training Dataset size

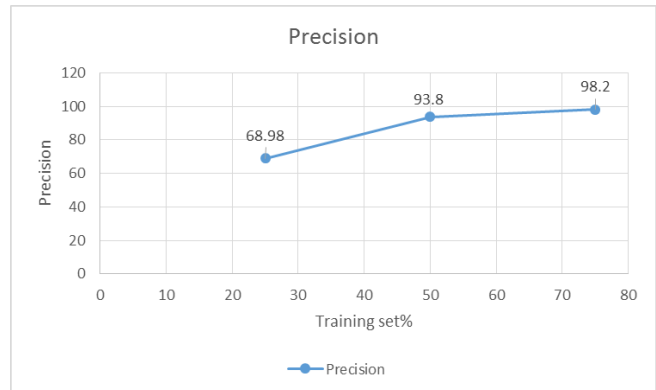


Fig 9.4 Precision VS Training Dataset size

### 9.2 Cloud Performance Graphs

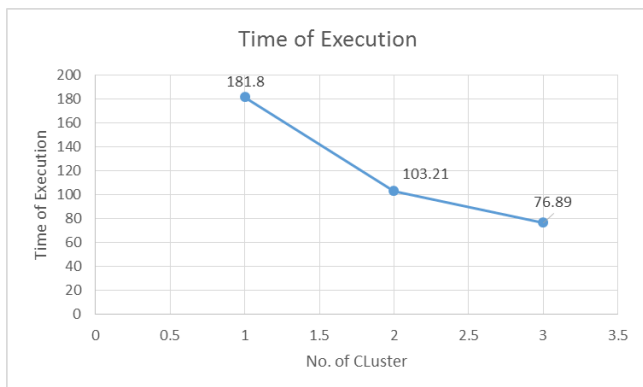


Fig 9.5. Time of Execution vs cluster size

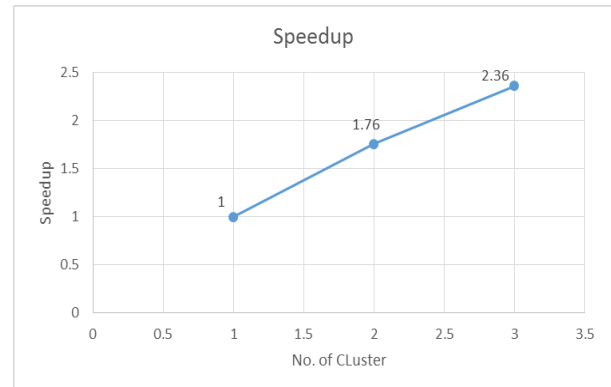


Fig 9.6. SpeedUp vs cluster size

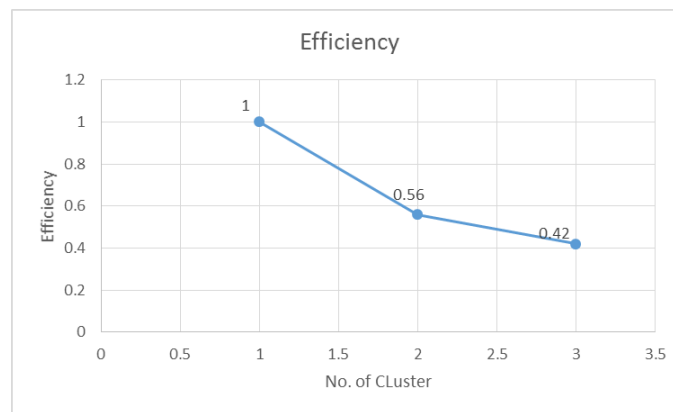


Fig 9.7 Efficiency vs Cluster size

### 9.3 Cloud Watch Graph from AWS

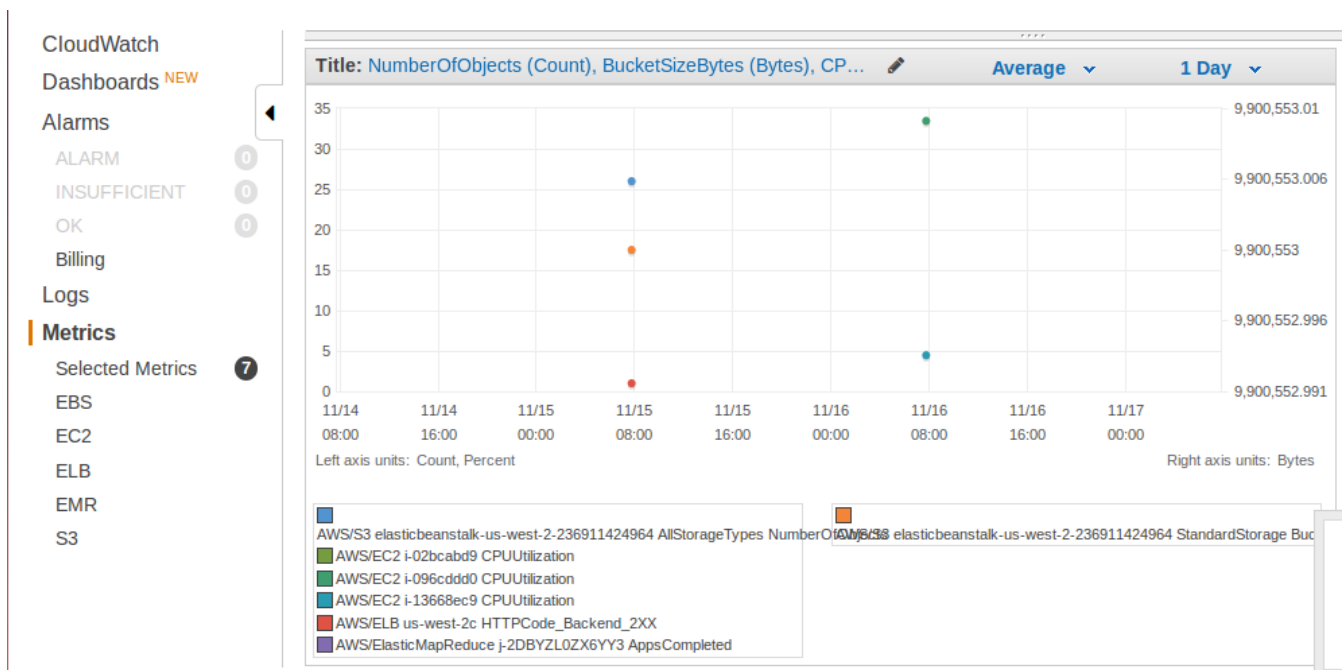


Fig 9.8 Performance metric monitor using AWS CloudWatch

## 10. ANALYSIS OF RESULTS

### 10.1 Machine learning performance:

In this section, we analyze the experimental results obtained by the python execution on amazon EC2 which evaluate the performance of the Naïve Bayesian (NB). There are four types of performance related to Machine learning technique. In this project, we have analyzed accuracy, F1 score and Recall and precision related to performance metrics. Following are the performance metrics observation:

- Accuracy increases with increase in Training Dataset
- F1 score increases with increase in Training Dataset
- Recall increases with increase in Training Dataset
- Precision increases with increase in Training Dataset

## 10.2 Cloud Performance

We deployed our naïve based filtering model on three different Hadoop cluster configurations of :

- 1 master 1 core
- 1 master 2 cores
- 1 master 3 cores

Note : instance type used is m3.xlarge

We monitored the performance of our model across these configuration and came with the performance comparison on the basis of speedup, time of execution and efficiency

- Execution time decreases with increase in cluster size
- Speedup increases with increase in cluster size
- Efficiency decreases with increase in cluster size

## 11. CONCLUSION

The analysis of the data should be scalable, flexible and high performance as the data increases in size. The Machine learning techniques are used for providing analysis and insight in a timely fashion. AWS gives many solutions for solving big data analytical problems. Most of the big data solutions use AWS services to meet business requirements in the most cost-optimized and resilient way possible. The result is a very flexible big data architecture which scales with our business on AWS global infrastructure.

In our project, we used Naïve based Machine Learning technique which is simple yet very powerful in classifying data. Naïve based is supervised Machine Learning Technique which assumes that the features are independent. The performance of Machine Learning increases with increases in training datasets.

## 12. REFERENCE

- [1] Seongwook Youn, Dennis McLeod "A Comparative Study for Email Classification", Advances and Innovations in Systems, Computing Sciences and Software Engineering, 2007
- [2] <https://aws.amazon.com/ec2/instance-types>
- [3] [www.python.org/about/](http://www.python.org/about/)
- [4] <https://aws.amazon.com/s3/storage-classes/>
- [5] 'HiBench: A Representative and Comprehensive Hadoop Benchmark Suite'- Shengsheng Huang, Jie Huang, Yan Liu, Lan Yi and Jinquan Dai, Intel Asia-Pacific Research and Development Ltd., Shanghai, P.R.China, 200241
- [6] 'The HiBench Benchmark Suite: Characterization of the MapReduce-Based Data Analysis', Shengsheng Huang, Jie Huang, Jinquan Dai, Tao Xie, and Bo Huang Intel China Software Center, Shanghai, P.R.China, 200241
- [7] [https://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)
- [8] A paper describing the Enron data was presented at the 2004 CEAS conference.
- [9] The original dataset downloaded from William Cohen's web page (<http://www-2.cs.cmu.edu/~enron/>)
- [10] <http://blogs.msdn.com/b/andreasderuiter/archive/2015/02/09/performance-measures-in-azure-ml-accuracy-precision-recall-and-f1-score.aspx>
- [11] <http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-what-is-emr.html>
- [12] Zend Whitepaper PHP - Zend Technologies Inc
- [13] <https://aws.amazon.com/cloudwatch/>