CSE 535 MOBILE COMPUTING
BenchMarking Machine Learning on mobile phone CPU
Arizona State University

Siddhant Sudan
1212890391
ssudan@asu.edu

Deepak Haldar
1211136508
Deepak.haldar@asu.edu

Abhinethri Tumkur Umesh
1213187714
atumkuru@asu.edu

Saptarshi
1213245863
Schowd12@asu.edu

ABSTRACT

In this project, we have developed an android application which benchmarks four Machine Learning(ML) algorithms on the spam base dataset. For benchmarking, we are calculating different measures namely True Accept Rate (TAR), True Reject Rate (TRR), False Accept Rate (FAR) and False Reject Rate (FRR). The four machine learning algorithms which our team has selected are Naïve Bayes Classifier, Support Vector Machine (SVM), Logistic Regression and Multilayer Perceptron. The Data Set has been chosen from UCI repository to classify the message as either spam or ham. We have used weka library for the implementation of the above-mentioned algorithms.

Author Keywords: SVM, logistic Regression, Naïve Bayes, Neural Network, Multilayer Perceptron, Weka, TAR, TRR, FAR, FRR , Train, Test

INTRODUCTION

Benchmarking with respect to machine learning means running different machine learning algorithms on the dataset and making comparisons among them which one performs better. Benchmarking compares different machine learing algorithms with respect to how well they can learn the patterns from the dataset. Benchmarking can be used as a sanity check to see that a new algorithm runs successfully and compares well with the existing algorithms. It basically measures the performance of an algorithm. Benchmark datasets can be of two types. First one is "real world dataset" which represents real word problems. Second is "simulated dataset" this is a dataset which is generated artificially to mimic real world data. We have considered four different algorithms for the purpose of comparing results. They are namely Support Vector Machine, Logistic Regression, Naïve Bayes classifier and Multilayer Perceptron. We run these algorithms on the dataset. The training part is done in the cloud and the model which is created is then sent to the smartphone where test data is present. The model is evaluated on the basis of how well it performs on the test data. We calculate different metrics for comparing the performance of the algorithm. The rest of the report consists of implementation, challenges, completion of task, conclusion, future work, references and acknowledgement.

IMPLEMENTATION

1.DataSet
The required DataSet to distinguish a message as spam or ham has been downloaded from UCI repository. The last column tells whether data is spam or not. The sample data has been collected from both office and personal mails. The algorithms Output has only 2 classes, yes for spam and no for ham.

2. Data Split
In the home page, user has an option to choose the train and test data. Split option has been provided along with the selection of an algorithm. Generally, small data is used for testing and remaining part is used as training data. Thus, user will always choose similar data for training and testing irrespective of the size to reduce the data discrepancies and to understand the ML algorithms performance.

3.Weka for implementation
The library is open source and has ML algorithms mainly for data mining tasks. We are using the classes and methods from Weka to evaluate the models based on the user inputs.

4. Flow of the Application
After user chooses an algorithm and data parameters corresponding to the chosen algorithm for the evaluation, TAR, TRR, FAR and HTER will be displayed along with the execution time and the information will be logged for the comparison.

5. Support Vector Machine(SVM)
The parameters considered for SVM are SVM type (C-Svc and nu-SVC), Kernel type, tolerance, degree, size, epsilon and gamma. Weka class used is SMO which implements sequential minimal optimization algorithm for training. The default values are displayed for the ease of use. The user can execute with the default input values or modify for the execution of the classifier.

6. Naïve Bayes
This is a probabilistic classifier and based on Bayesian theorem, it has 2 important parameters. The 1st parameter is Kernel density and suppose the input for this parameter is 0.1, the values in the interval (0.55,0.65] will all be treated as 0.6. The second parameter is Discretization which can be either supervised or unsupervised. Weka class name is NaiveBayes.

7. Logistic Regression
The parameters considered for this classifier are ridge and maximum number of iterations. The number of iteration should be in the range of 0-100, optimum value is 50. The class logistic along with the ridge estimator has been used for the implementation.

8. Multilayer Perceptron
MultilayerPerceptron class of Weka has been used for implementation, it uses backpropagation for the classification. There are several parameters for this classifier. After thorough researcher, we have considered learning rate, momentum, size of validation set, epoch, and threshold for number of consecutive errors. The default values for all the parameters have been provided for the ease of use.

9. Cloud Implementation
Amazon WebService(AWS) has been used for cloud part as it is reliable and fast. The 2 services which has been mainly used are lambda and S3. S3 has been used to store the created objects and dataset of .arff format (S3 expects the objects to be in the form of file). Lamda has been used for logic implantation in Java8. Whenever a model gets evaluated corresponding object will be created in S3.

10. Logfile
Whenever an algorithm runs, the results will be updated in the log. The log has the following information: data-split information, classifiers, time taken and other related information. A user can refer this for comparing the performance. The log file can be accessed from the home page. The information will be appended to the existing log.

11. Testing
Each of the team members worked exclusively on one algorithm and tested the evaluation of other 2 algorithms and 4th algorithm i.e. logistic regression was implemented and tested by all the members. For example, Team member who worked on SVM, tested multilayer perceptron and Naïve Bayes which were developed by other two members.

12. Integration
Since each team member worked on different classifiers, we dedicated few hours for deciding the flow, User Interface(UI) for home page, access and display of log files. Along with concentrating on the correct implantation of classifiers, we also spent time on improvising the UI with proper alignment for better user experience.

CHALLENGES

1.Training the model with the parameters on the cloud.
2.Calculating the response time of the overall cloud computation.
3.Paramater parsing from the application to the cloud.
4.Object formatting-had to convert machine learning object to file object for downloading and uploading.
5.Multithreading (to run each algorithm on a separate thread without affecting the performance of the algorithm).

COMPLETION OF TASKS

No.	Tasks	Member Name
1	Implementation of Support Vector Machine algorithm by finding appropriate methods and classes	Deepak Haldar
2	Connecting to the cloud to train using Support Vector Machine and fetching the results	Deepak Haldar
3	GUI for SVM pages	Deepak Haldar
4	Selecting the appropriate parameters out of available parameters as inputs for SVM	Deepak Haldar
5	Researching on the input values for optimized results and setting them as default values	Deepak Haldar
6	Integration of the entire application and making the video	Deepak Haldar
7	Testing Naïve Bayes and Multilayer Perceptron results	Deepak Haldar
8	Implementation of Naïve Bayes Classifier by finding the appropriate classes and methods	Abhinethri Tumkur Umesh
9	Connecting to the cloud to train using Naïve Bayes and fetching the results	Abhinethri Tumkur Umesh
10	GUI for Naïve Bayes Classifier pages	Abhinethri Tumkur Umesh
11	Implementation of log files	Abhinethri Tumkur Umesh
12	Selecting the appropriate parameters out of available parameters as inputs for NaiveBayes	Abhinethri Tumkur Umesh
13	Researching on the input values for optimized results and setting them as default values	Abhinethri Tumkur Umesh
14	Testing SVM and Multilayer Perceptron results	Abhinethri Tumkur Umesh
15	Implementation of Multilayer by finding the appropriate classes and methods	Siddhant Sudan

16	Connecting to the cloud to train using Multilayer Perceptron and fetching the results	Siddhant Sudan
17	GUI for Multilayer Perceptron pages	Siddhant Sudan
18	Selecting the appropriate parameters out of available parameters as inputs for Multilayer Perceptron	Siddhant Sudan
19	Researching on the input values for optimized results and setting them as default values	Siddhant Sudan
20	Implementation of UI for home page and Setting the environment for the development	Siddhant Sudan
21	Testing Naïve Bayes and SVM results	Siddhant Sudan
22	Implementation of Logistic Regression completely	Abhinethri, Deepak and Siddhant
23	UI for Displaying logs	Saptarshi

CONCLUSION

As per the analysis done on all the algorithms we found out that Multilayer perceptron gave better results as compared to other algorithms. Due to the no of hidden layers and complexity of the neural network this algorithm performed better. Through this project we came to know a lot about the implementation a machine learning algorithm,weka library and impact of machine learning on mobile computing.

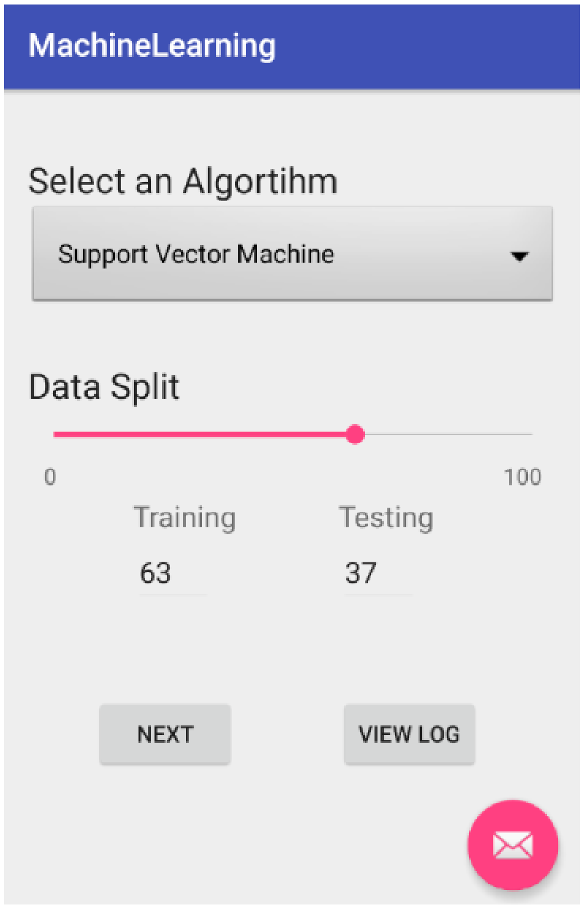
FUTURE WORK

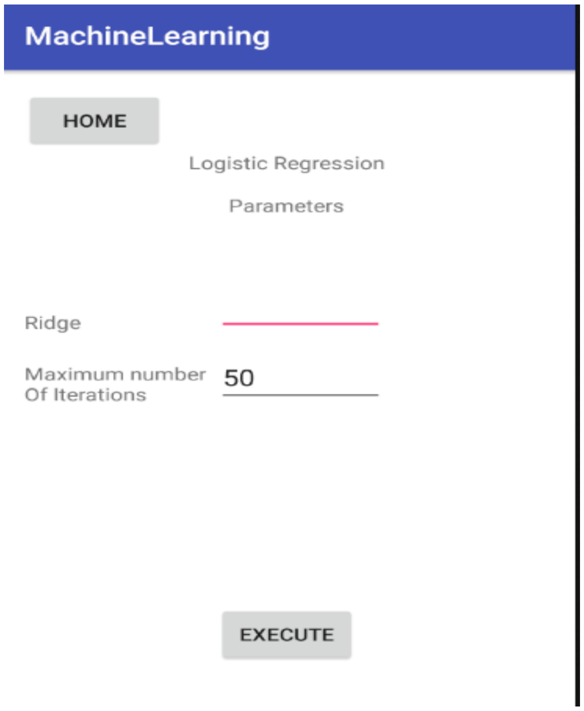
In the future we can include more algorithms instead of just four so that we can make exhaustive comparisons among them. This can drastically improve our analysis.We can also take large datasets which have millions of records so that the learning phase improves and we get a more robust model. We can also perform Benchmarking on GPU to see how it performs with respect to CPU.We can use different datasets for comparison to see whether any algorithm performs well on a specific type of dataset.

ACKNOWLEDGEMENT

I would like to thank Ayan Banerjee for giving us an opportunity to learn Android and Machine Learning Concepts in a short span. The implementation of all the four algorithms wouldn't have been possible without his guidance.

SCREENSHOTS





MachineLearning

HOME

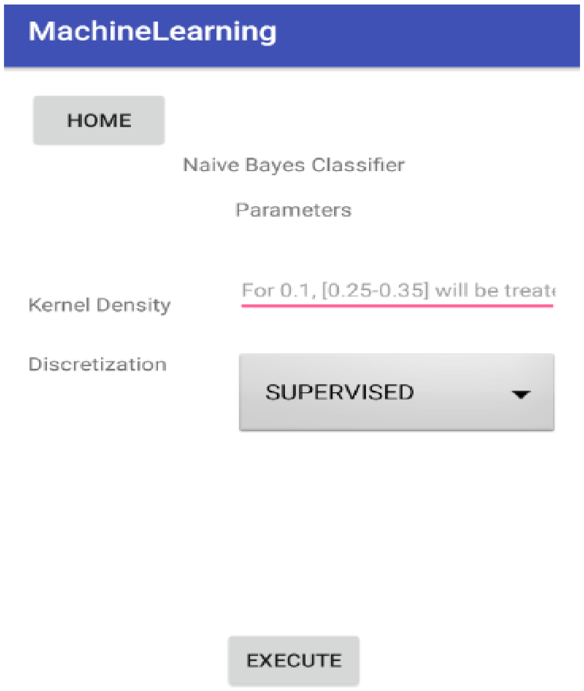
Logistic Regression

Parameters

Ridge

Maximum number Of Iterations 50

EXECUTE



MachineLearning

HOME

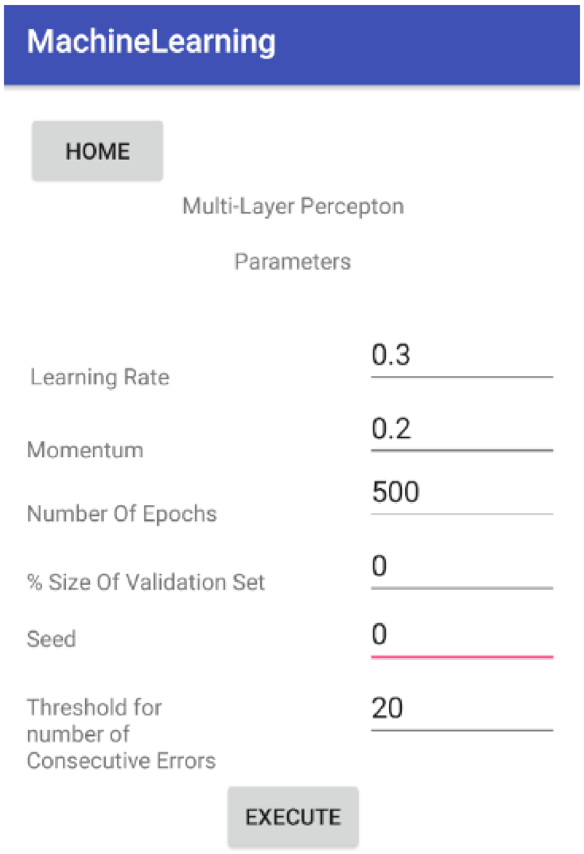
Naive Bayes Classifier

Parameters

Kernel Density For 0.1, [0.25-0.35] will be treat

Discretization SUPERVISED

EXECUTE



MachineLearning

HOME

Multi-Layer Percepton

Parameters

Learning Rate 0.3

Momentum 0.2

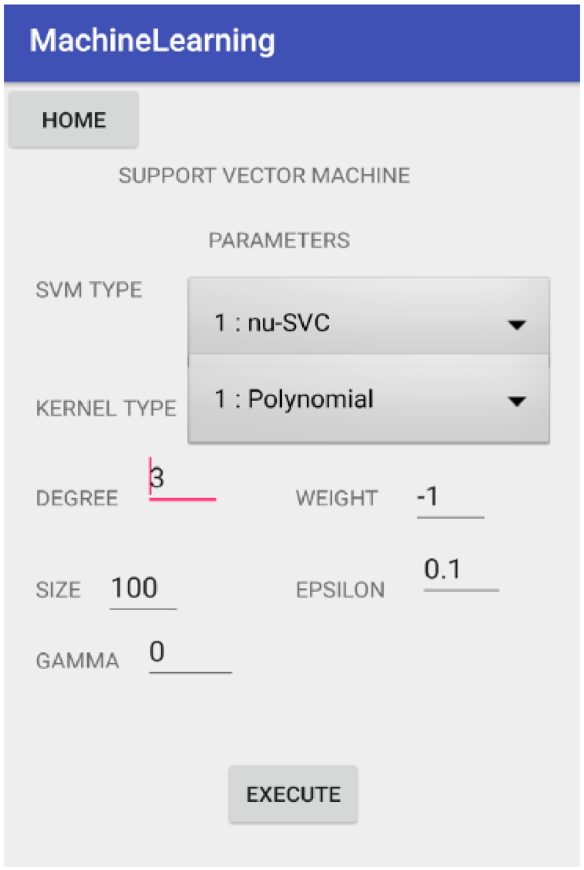
Number Of Epochs 500

% Size Of Validation Set 0

Seed 0

Threshold for number of Consecutive Errors 20

EXECUTE



MachineLearning

HOME

SUPPORT VECTOR MACHINE

PARAMETERS

SVM TYPE 1 : nu-SVC

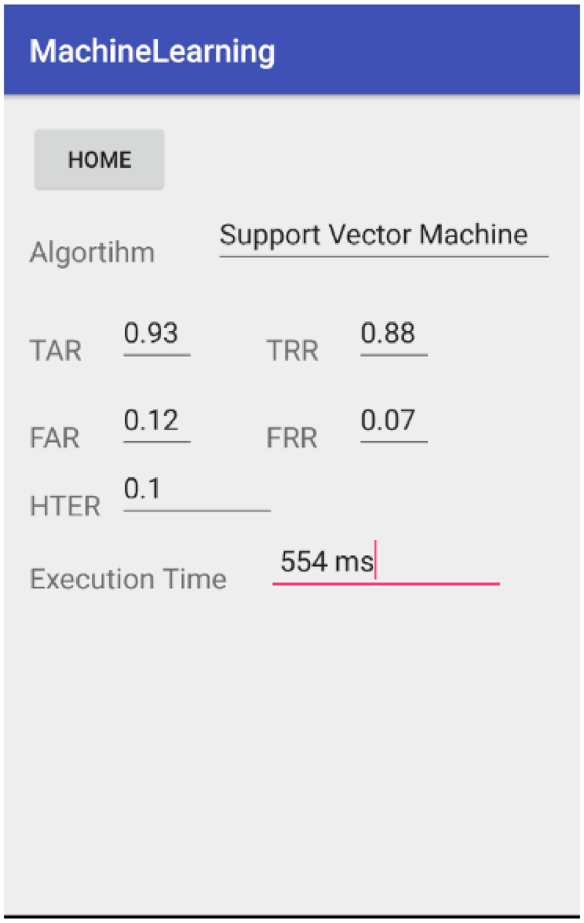
KERNEL TYPE 1 : Polynomial

DEGREE 3 WEIGHT -1

SIZE 100 EPSILON 0.1

GAMMA 0

EXECUTE


MachineLearning

HOME

Algortihm	Support Vector Machine		
TAR	0.93	TRR	0.88
FAR	0.12	FRR	0.07
HTER	0.1		
Execution Time	554 ms		

REFERNCES

1.SVM
http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html

2.Naïve Bayes
http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayes.html

3.Multilayer Perceptron
http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/MultilayerPerceptron.html

4.Logistic Regression
http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/Logistic.html

5. Cloud Implementation

6. Machine learning algorithms in Weka
https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/

7.Weka on Android
http://www1.cuni.cz/~obo/vyuka/projekty/figura-ml-for-android.pdf

8.ML for Java Developers
https://www.javaworld.com/article/3224505/application-development/machine-learning-for-java-developers.html

9.Developemnt of Android App
https://developer.android.com/training/index.html

11. Reading and writing to a file(Logs)
https://www.youtube.com/watch?v=CNoj9vzAYiQ

12. DataSet
https://archive.ics.uci.edu/ml/datasets/spambase