PERFORMANCE EVALUATION OF DATA ANALYTICS TECHNIQUES ON DIFFERENT CLOUD PLATFORMS

Stefan D'Costa
School of Engineering
Santa Clara University
California, USA

Clarissa Sequeira
School of Engineering
Santa Clara University
California, USA

Prarthana Raghavan
School of Engineering
Santa Clara University
California, USA

Amy Truong
School of Engineering
Santa Clara University
California, USA

***Abstract*— Major cloud platforms provide many data analytics solutions for its users and this paper aims at finding out which cloud platform performs better for which data analytical technique. This paper compares Microsoft Azure, Google Compute Engine and Amazon Web Services cloud platforms using the data analytics techniques like Segmentation, Logistic Regression, Text Mining and Clustering. The comparison will be done based on the time taken to execute each technique, the cost of service, service level agreements, and the infrastructure provided. The implementation of these data analytics algorithms is done in R programming language. This comparison will summarize the performance of different cloud platforms and help in choosing best cloud platform for a particular data analysis technique.**

Keywords—Cloud Platforms; Clustering; Regression; Text Mining; Segmentation; R programming;

I. INTRODUCTION

For our research project, we evaluated the performance of four different data analytic techniques on three cloud platforms. The three cloud platforms we examined were Google Compute Engine, Microsoft Azure, and Amazon Web Services. As for the different data analytic techniques, we chose to test: Clustering, Logistic Regression, Text Mining, and Segmentation. Our evaluation looked at four quality of service parameters which were time, cost, infrastructure, and service level agreements (SLAs) of each cloud platform.

This research topic is important because it combines two popular and emerging topics: cloud computing and data analysis. Cloud computing is a popular and commonly used technology perform data analysis – a process that provides insightful information on large datasets. Data analysis requires powerful techniques and platforms to perform efficiently. We felt it is important to find out which data analytic technique performs the best on which cloud platform to improve the process of data analysis.

Currently other solutions, which are online articles or company marketing, focus on comparing costs and list the data analytic techniques provided by each cloud platform, but not a comparison of how well the techniques perform. This information does little to help users quickly decide which technique or platform to use if they are limited with their time. Also, other factors that these online articles fail to take into account are time and reliability of the cloud infrastructure.

Our results:

- Show which data analysis technique is best used on which cloud platform to solve data analysis related problems.
- Show which cloud platform is the most cost-efficient, time-efficient, reliable – infrastructure wise.

Our results are important, because it will help users decide quickly which data analysis technique and cloud platform are best used together when they are trying to analyze their data. Users need to solve problems quickly and do not have time to test their technique on each cloud platform to find out which one performs the best.

II. LIST OF CONTRIBUTION

The main contribution of this paper is providing knowledge on measuring the performance of the commercial Cloud Platforms based on time, cost, service level agreements and infrastructure provided. This paper tries to achieve a knowledge stand on three main factors on a cloud based on:

A. Analytical processes

Dealing with simple analytical programs which are being used in the current age in the industry in order to establish patterns of studies on data.

B. Using a non-scalable language

Primary goal of making use of a non-scalable language like R is to ensure that the stress of performance is concentrated more on the analytical methodology used for the dataset and the performance of the same.

C. Infrastructure and SLAs

This paper aims at comparing the infrastructure metrics and service level agreements commitments of different cloud platforms.

Having referred various sources quotes on the internet and in publications, each of which point towards a more cost oriented analysis every time. We attempted to make a covered analysis of three parameters, based on each style of the processes to measure the corresponding performance/efficiency.

GitHub Repo: https://github.com/ClarissaSequeira/Cloudings.git

III. Related Work

Due the course of this research two IEEE papers stood out as these papers too tried to achieve a comparison between cloud platforms.

The paper titled "Understanding the Performance and Potential of Cloud Computing for Scientific" does a performance evaluation of the scientific aspects of the AWS cloud first by evaluating the raw performance of different services of AWS such as compute, memory, network and I/O. And then, based on the collective results of these parameters, it gauges the cumulative limitations on the scientific applications and what causes the setbacks on the cloud. This in turn, is compared to the Private Cloud operations of the AWS, so that it will highlight the functionality that results in lack of performance in scientific operations over the private cloud ones.

The paper titled "Performance evaluation of Cloud Service Providers" primarily compares the CSPs (Cloud Service providers) by means of gauging the performance of the commercial CSPs on seven different metrics. However, this analysis was performed on CSPs not included in our project.

IV. Data Analytic Techniques

A. Segmentation

Segmentation is a data analytic technique where data is grouped based on similarities or *dissimilarities.* In the case of segmentation, the categorization of the information in a standardized structured format. [1] In order to achieve this end result algorithm perform the process of narrowing down by two major means: Agglomerative method and Divisive method (used in this paper).

The package used in R for achieving segmentation: NMF

The prime function used is: nmf (x, rank, method, seed, ...)

x	<- target matrix, data.frame or ExpressionSet
rank	<- factorization rank, i.e. the number of columns in the resultant matrix
method	<- is the algorithm used to estimate the factorization
seed	<- is the seeding method used to compute the starting point.

Dataset Size used: 50 MB
Rows: 8400
Variables: 7
Execution Time for AWS: 28.18 s
Execution Time for Azure: 29.58 s
Execution Time for Google Compute Engine: 30.036 s
Result: Sorted order of brands which were preferred by customers and showed how many customers given a brand would buy it based on a factor like (free gift on purchase).

Challenges Faced:
(i) The packages required for segmentation was difficult to load on cloud platforms.
(ii) The normalization of the dataset has to be done based on the best-case selection of the column that will be unique and useful in deducing patterns

B. Logistic Regression

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous (binary) variable, i.e. it only contains data coded as 1 (TRUE, success, pregnant, etc.) or 0 (FALSE, failure, non-pregnant, etc.) [2]. Logistic Regression is based on the logit function as shown below:

logit(p) = ln (p/1-p)

p	<- probability of an event occurring
1-p	<- probability of an event not occurring
p/1-p	<- odds of an event occurring
logit(p)	<- logged odds

Generalized Linear Model (glm) function is used to find the best-fit regression line with maximum likelihood estimation.

glm (formula, data = titanic_dataset, family=binomial (link='logit'))

formula	<- a symbolic description of the model to be fitted
family	<- a description of the error distribution and link function to be used in the model
data	<- an optional data frame, list or environment containing the variables in the model

Dataset Size used: 60 MB
Rows: 953,634
Variables: 8
Execution Time for AWS: 21.23 s
Execution Time for Azure: 10.69 s
Execution Time for Google Compute Engine: 17.55 s
Result: Predicted the probability of survival of each passenger on board the Titanic with an accuracy of 80%.

Challenges Faced:
(i) Loading the entire dataset on the cloud platforms.
(ii) Performing missing value and outlier treatment.

C. Text Mining

Text mining, a specific technique within the data mining field, discovers insightful information by extracting data from data sets. It discovers information unknown to the user, unlike web searching, by extracting from natural languages. Examples of different applications include something as simple as finding the most common words in a book, to summarizing articles, extract resume attributes to fill an online application, or reading tweets to discover trending topics.

The experiment ran an R script using the text mining, "tm", library.

Dataset Size used: 64 MB
Execution Time for AWS: 27.99 s
Execution Time for Azure: 24.45 s
Execution Time for Google Compute Engine: 32.71 s
Result: Found the most common words in all of South Park scripts

D. Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). [4]

k-means clustering aims at partitioning 'n' observations into 'k' clusters in which each observation belongs to a cluster with the nearest mean, serving as a prototype of the cluster.

The function used for k-means clustering in R is:

kmeans (dataset, no)

dataset	<- clean dataset matrix
no	<- number of clusters

Dataset Size used: (wine.csv) 200 MB
Rows: 3,225,360
Variables: 14
Execution Time for AWS: 30.6 s
Execution Time for Azure: 24.5 s
Execution Time for Google Compute Engine: 41.4 s
Result: Divided the dataset into in 3 clusters based on closed Euclidean distance between data points.

Challenges Faced:
(i) Loading the dataset on all the cloud platforms was a tedious and time consuming task.
(ii) Getting a constant execution running time. This was achieved by increasing the dataset size.

V. Problem Analysis – Parameters for Comparison

The parameters used to address the question explained in the introduction are as follows:

A. *Time Analysis*

This proposed experiment will capture the running time of each data analytical technique in milliseconds (ms) on each cloud platform.

B. *Cost Analysis*

The experiment will give an approximate cost in USD ($) for each data analytical technique on each cloud platform.

C. *Infrastructure Provided*

The experiment will also document the default infrastructure provided for each cloud platform. Infrastructure parameters include CPU, RAM and Clock Speed.

D. *Service Level Agreements (SLAs) Provided*

This paper will also record the different SLAs provided by each cloud platform and provide an insight to the security levels, commitments and deliverables for each cloud platform.

VI. Evaluation Methodology

The evaluation implementation is mentioned in the following steps:

Step 1: Create an account with all three cloud service providers. AWS – amazon account (account liked to Amazon.com). Azure – Microsoft account (Hotmail /Outlook). Google Compute Engine – Google account (Gmail).

Step 2: Edit the security groups to open port 8787 to connect to RStudio server.

Step 3: Launch the instances and open terminals for each cloud instance.

Step 4: Install R libraries, RStudio and RStudio server on each instance.

Step 5: Use the public IP address and port number 8787 to connect to RStudio server and login with credentials.

Step 6: Import R script and dataset from local machine to each cloud instance and load the appropriate packages (if required).

Step 7: Run the R script and document run time for ten executions and document the results.

Step 8: Evaluate the results with cost and infrastructure.

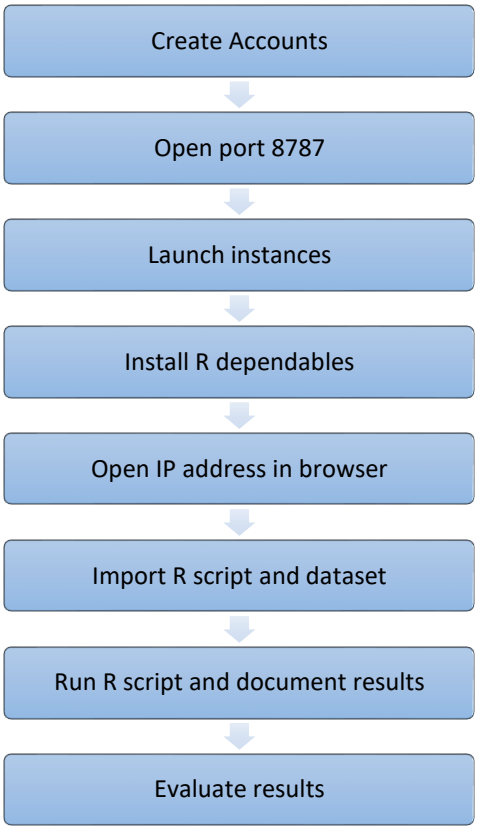

Figure1: Flowchart of the proposed implementation.

VII. Observations and Graphs

Below are the results obtained for the execution runs on different cloud platforms.

For logistic Regression:

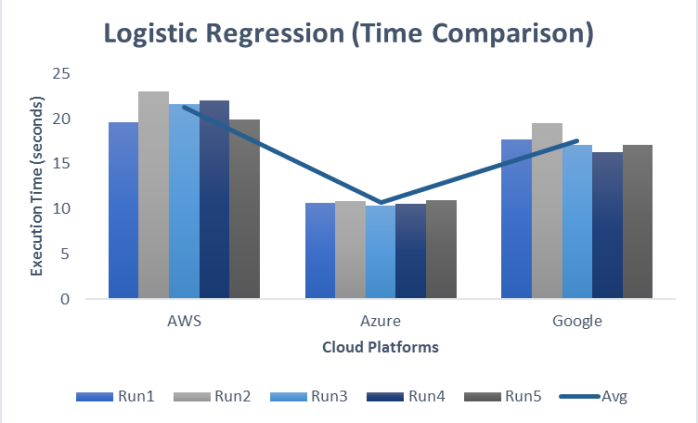

Figure 2: Average Execution time for each platform using minimum 5 executions. (logistic regression)

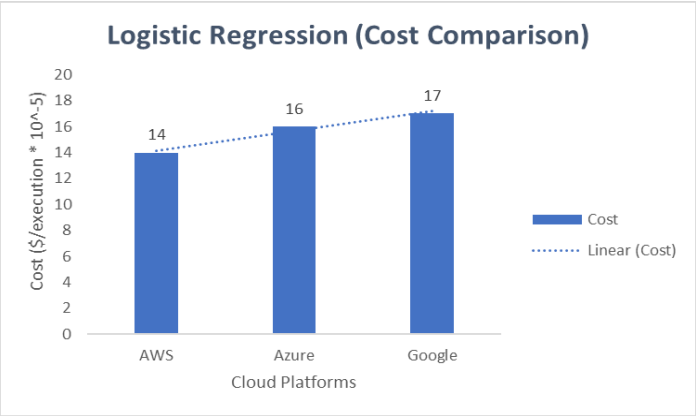

Figure 3: Cost Analysis (logistic regression)

For Segmentation:

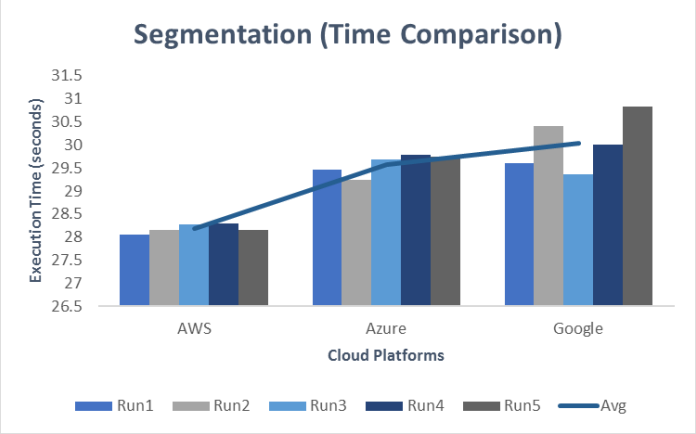

Figure 4: Average Execution time for each platform using minimum 5 executions. (segmentation)

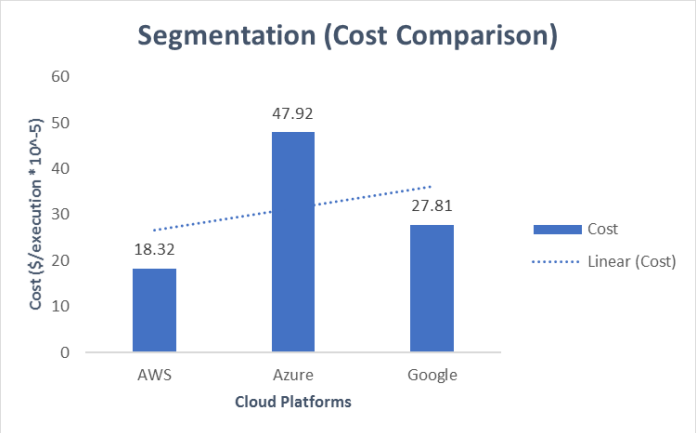

Figure 5: Cost Analysis (segmentation)

For Clustering:

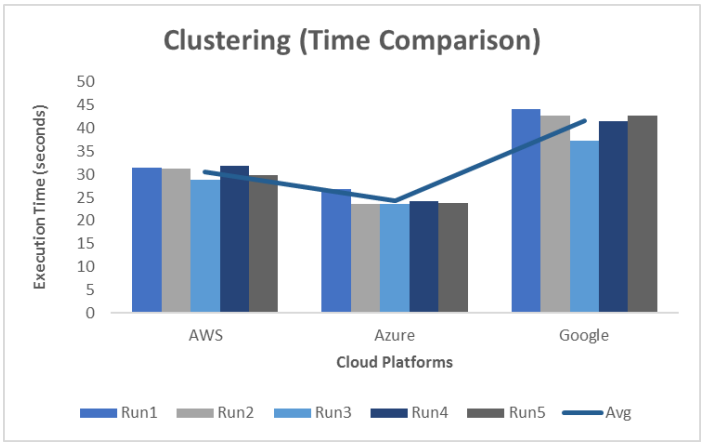

Figure 6: Average Execution time for each platform using minimum 5 executions. (clustering)

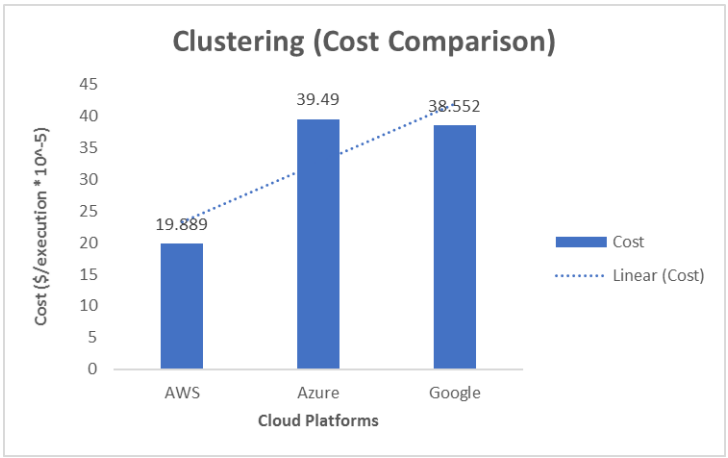

Figure 7: Cost Analysis (clustering)

For Text Mining:

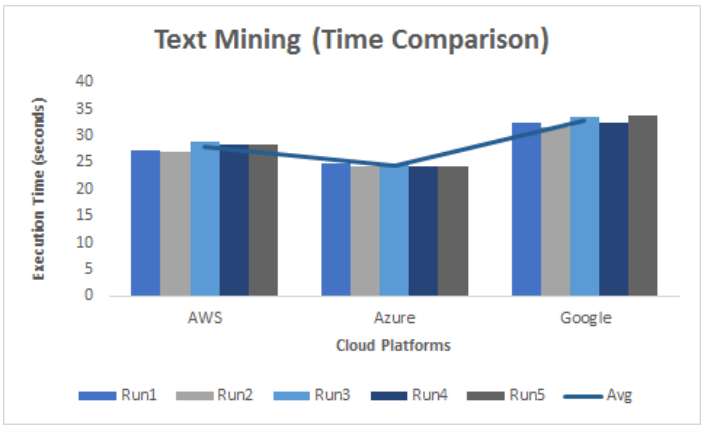

Figure 8: Average Execution time for each platform using minimum 5 executions. (text mining)

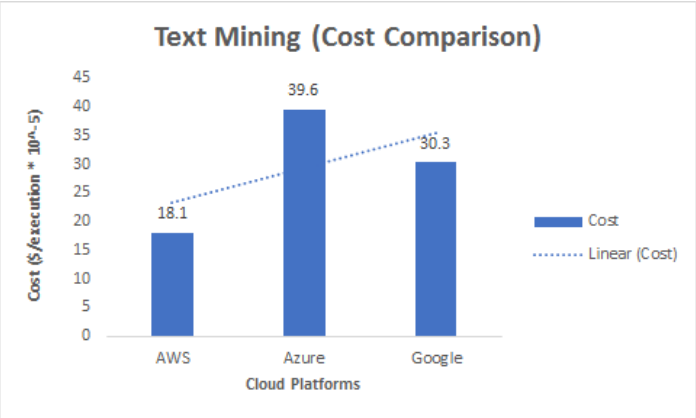

Figure 9: Cost Analysis (Text Mining)

VIII. Summary and Conclusion

Below are the Summary tables of all our findings:

Technique	Time Analysis
Regression	1. AWS (21.23s) 2. Azure (10.69s) 3. Google (17.55s)
Clustering	1. Azure (24.38 s) 2. AWS (30.59 s) 3. Google (41.63s)
Segmentation	1. AWS (28.19 s) 2. Azure (29.583s) 3. Google (30.04 s)
Text Mining	1. Azure (24.45 s) 2. AWS (27.99 s) 3. Google (32.79 s)

Table 1: Time Analysis Summary

Technique	**Cost Analysis (Rate*Time)** **Rate:** **AWS ($6.5*10^-6 / sec)** **Azure ($16.2*10^-6 / sec)** **Google ($9.26*10^-6 / sec)**
Regression	1. AWS ($1.4*10^-4 / execution) 2. Google ($1.62*10^-4 / execution) 3. Azure ($1.73*10^-4 / execution)
Clustering	1. AWS ($1.98*10^-4 / execution) 2. Google ($3.85*10^-4 / execution) 3. Azure ($3.95*10^-4 / execution)
Segmentation	1. AWS ($1.83*10^-4 / execution) 2. Google ($2.70*10^-4 /execution) 3. Azure ($4.7*10^-4 / execution)
Text Mining	1. AWS ($1.81*10^-4 / execution) 2. Google ($3.03*10^-4 / execution) 3. Azure ($3.96*10^-4 / execution)

Table 2: Cost Analysis Summary

Technique	**Infrastructure Used**
Regression Clustering Segmentation Text Mining	AWS - Ubuntu 16.04 LTS 64 bit (AMD64 Xenial) Azure - Ubuntu 16.04 LTS 64 bit (AMD64 Xenial) Google - Ubuntu 16.04 LTS 64 bit (AMD64 Xenial)

Table 3: Infrastructure Summary

Technique	**Service Level Agreements**
Regression Clustering Segmentation Text Mining	AWS (Free up to 1CPU and 1GB RAM and for 1 CPU with 2GB RAM they charge $16.84 per month) 2. Azure (Any service with $200 credit up to 1 month and $42 per month for 1CPU and 3.75GB RAM) 3. Google (Any service with $300 credit up to 1 year and $24 per month for 1 CPU and 3.75 GB RAM)

Table 4: Service Level Agreements

The findings of this paper do not aim to be biased towards any cloud service provider but just gives a comparative study of the cloud platforms on these particular techniques.

The findings indicate that execution wise AWS and Azure performed the best whereas Google was not up to the mark.

Cost wise per execution and storage AWS and Google provided the most cost efficient whereas Azure was very expensive.

Infrastructure wise all the four techniques used Ubuntu 16.04 LTS operating system with 64 bit AMD Xenial processor.

Based on experience Google had the most user friendly experience whereas operations and utility wise AWS was the best. Microsoft Azure was a little difficult to use and a confusing console.

IX. Acknowledgements

A special thanks to Louis Aslett for providing a readymade AMI for RStudio server.

A big thanks goes to Prof. Dr. Abhishek Gupta for his mentorship and guidance throughout this paper.

Thanks to RStudio and Kris Eberwein for documenting the steps to install RStuido on Linux OS.

X. Limitations

The cloud platforms instances were limited to the free tier available for each cloud platform.

The execution time for each process were the average of five consecutive executions and were not constant for each execution.

For full-fledged data analytics applications involving networking these observations may not hold.

XI. Future Scope

Future implementation could extend towards other data analytical techniques like MapReduce, Decision Trees, Support Vector Machines, Random Forests and others.

The comparison could also include other implementations on Docker engine and include other metrics like Network Latency and others.

References

[1] MEDCALC: easy-to-use statistical software [Online]. Available: https://www.medcalc.org/manual/logistic_regression.php

[2] B. Posey (2015). Compare the market-leading public cloud providers [Online]. Available: http://searchcloudcomputing.techtarget.com/feature/Compare-the-market-leading-public-cloud-providers

[3] Wikipedia Cluster Analysis [Online]. Available: https://en.wikipedia.org/wiki/Cluster_analysis

[4] R Documentation. Fitting Generalized Linear Models [Online]. Available: https://stat.ethz.ch/R-manual/R-patched/library/stats/html/glm.htm

[5] I. Sadooghi, J.H. Martin and T. Li. Understanding the Performance and Potential of Cloud Computing for Scientific Applications [Online]. Available: http://ieeexplore.ieee.org/document/7045591/

[6] B. E. Zant and M. Gagnaire. Performance evaluation of Cloud Service Providers [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7156482/authors

[7] Basic Text Mining in R [Online] Available: https://rstudio-pubs-static.s3.amazonaws.com/31867_8236987cf0a8444e962ccd2aec46d9c3.html#just-the-basics.

[8] Market Segmentation [Online] Available: https://ds4ci.files.wordpress.com/2013/09/user08_jimp_custseg_revnov08.pdf

[9] Customer Segmentation [Online] Available: https://www.slideshare.net/JimPorzak/user2015-jporzak-customersegmentation20150701.

[10] Louis Aslett [Online] Available: http://www.louisaslett.com/RStudio_AMI/

[11] Abhishek Gupta. The Who, What, Why, and How of High Performance Computing in the Cloud. [Online]. Available - http://charm.cs.illinois.edu/newPapers/13-30/paper.pdf

[12] Kris Eberwein. How to install R in Linux Ubuntu 14.04 [Online]. Available - https://www.datasciencecriot.com/33/kris/

[13] RStudio [Online]. Available - https://www.rstudio.com/products/rstudio/download-server/