https://doi.org/10.48047/AFJBS.6.14.2024.2536-2554

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

STUDENT'S PERFORMANCE PREDICTION USING HYBRID OPTIMIZATION ALGORITHM-BASED MAP-REDUCE FRAMEWORK

Mr. Mahendra Sawane, Dr. Pratap Singh Patwal

Research Scholar, Computer Science & Engineering
Nirwan University Jaipur Rajasthan
mahendra.sawane@nirwanuniversity.ac.in
Associate Professor, Computer Science & Engineering
Nirwan University Jaipur Rajasthan
faculty.engg07@nirwanuniversity.ac.in

Volume 6, Issue 14, Aug 2024

Received: 09 June 2024

Accepted: 19 July 2024

Published: 08 Aug 2024

doi: 10.48047/AFJBS.6.14.2024.2536-2554

Abstract:
An essential challenge in the field of education is predicting student performance since it enables teachers to spot pupils who could struggle academically. Machine learning makes extensive use of hybrid optimization methods to boost the precision of prediction models. Large datasets may be processed in parallel using the Map-Reduce programming style, which makes it the perfect platform for big data performance prediction applications. The prediction problem is split into two parts in a map-reduce architecture using a hybrid optimization technique. The data is split up and spread among many cluster machines in the initial stage. Each division is subjected to a machine learning method via the map function, which produces a series of intermediate outcomes. The reduction function integrates the intermediate outcomes in the second phase to create the final forecast.
The hybrid optimization algorithm combines two or more optimization techniques to improve the accuracy of the predictive model. For example, it may combine a genetic algorithm with a gradient descent algorithm to search for optimal values of model parameters. The optimization algorithm is applied to the training data to learn the parameters of the predictive model. The map-reduce framework provides a scalable and fault-tolerant infrastructure for processing large datasets. It can handle data that is too big to fit into the memory of a single machine by dividing the data into smaller chunks and processing them in parallel. This allows the prediction task to be completed faster than if it were processed on a single machine. Overall, a hybrid optimization algorithm-based map-reduce framework is a powerful tool for student performance prediction. By leveraging the scalability and fault-tolerance of the map-reduce framework and the accuracy of the hybrid optimization algorithm, educators can build predictive models that are both accurate and scalable to handle large volumes of data.
Keywords: Prediction algorithm, MapReduce, Distributed File System, Hadoop, Machine Learning

1. Literature Review

Many studies on the evaluation of different frameworks for their performance determination have been done. Researchers in Spain have tried to determine the performance comparison of frameworks like Hadoop and Spark in processing Big Data. In a research paper [1] researchers have focused on performance optimization with the replacement of the Hadoop framework with Spark framework. They concluded that the execution time could be improved by 77% in the case of the Spark framework for the non-sort benchmark. Extensive research [1] [2] [3] on traditional algorithms, implementation, and improvement within MapReduce has been done. Some researchers [1] [2] have proposed techniques of implementation for indexing and clustering algorithms in MapReduce frameworks. In a research paper titled "Map, reduce and MapReduce, the skeleton way." [2] The researcher has focused on the implementation of three algorithmic skeletons inside commercial off-the-shelf clusters. They have focused on the implementation aspect of the algorithm and not its algorithmic behaviour. There is scope for evaluation of algorithmic behaviour concerning factors impacting the performance of the MapReduce. Further, there has been a possibility for evaluation of the performance of the MapReduce programming model in different algorithms.

2. MapReduce programming model and Distributed File System

MapReduce concept was designed by Google Inc [19]. It has used various Google products like Google Search, Gmail, Google Plus and other Google products. A Big Data programming approach on commodity hardware is Google MapReduce. It provides Big Data computations for various usages. The architecture of the Google MapReduce consists of two functions. These functions are the Map stage and Reduce stage. Google MapReduce has been implemented in Google App Engine using Java or Python language. Google MapReduce is not an open-source product, so not much research is available in the open domain literature.

Data is stored on a network of systems using the distributed file system, a decentralized type of file system. These distributed file systems use client-server architecture and master-slave architecture. For data retrieval, the client nodes and resource managers make calls to the real data storage locations. Hadoop Distributed File System, Andrew File System, and Google File System are the most popular distributed file systems.
A distributed file system called Google File System was created by researchers at Google Inc. It was created with high failure tolerance for deployment on common hardware. As seen in Figure 2.1, it has three design objectives. Scalability, dependability, and availability are these objectives [13].

The Hadoop MapReduce is a part of Hadoop framework as programming model [16]. The map and reduce steps make up the Hadoop MapReduce process. Moreover, there is a stage between shuffling and sorting. It is a distributed computing paradigm that drew inspiration from Google MapReduce. Data of a very large size could be processed in a parallel manner like in a traditional system. Nevertheless, the speed of data processing remains within expected limits in MapReduce. It is due to parallel processing the data and distribution of the

complex problems to various Hadoop clusters. The Hadoop framework that allows users to focus on core business logic had automated all the data processing in distributed environments. It achieves the parallel processing by dividing the problem into small subsets then allocating in the distributed map and reduces workers. Lisp language had inspired and acted as a base for mapping and reduces concepts [19] construction.

The Hadoop framework includes a distributed file system called HDFS [16]. It is an essential part of the Hadoop system. The GFS served as a model for this distributed file system. It had been designed for handling large files with aims to reduce network-based workload (input and output procedures). The characteristics of HDFS are scalability and availability. Both of them had been possible due to the replication of data at multiple nodes. Replication provides fault tolerances in case of failure of any node it could be replaced by its replica node. Replication of the data had been self-governing in the Hadoop framework.

2.1. Model Architecture

2.1.1. Components of Mapreduce Architecture:

- **Client:** Jobs are submitted to MapReduce for processing by MapReduce clients. Various clients may be available that constantly send errands to Hadoop MapReduce Director for processing.
- **Job:** The actual work mentioned by the client was a MapReduce job consisting of a few more modest errands that the client wanted to work on or do.
- **Hadoop MapReduce Master:** Separate each job into successive job parts.
- **Job-Parts:** The work or undertakings that come about because of separating the essential job. The final result created when all the job-parts are coordinated.
- **Input Data:** The data set that MapReduce uses to process data.
- **Output Data:** The processing results in the end product.

Integrate the client into MapReduce. A company of a certain size is sent from the client to Hadoop MapReduce Expert. MapReduce ace now further divides this work into virtually identical job parts. The Lead and Accepting Company will work on these portions of the order at this time. This management/acceptance document contains products for understanding the details of the use cases handled by each company. Designers justify to meet the details set by the region. Leadership tasks are provided with informational data to use, and leaders deliver rational key-benefit pairs as a result. The Minimizer gets the result of the Guide, or these key-esteem coordinates, and stores the outcome on the HDFS. It is feasible to make n different Guide and Lessen undertakings to deal with the data on a case by

case basis. The Guide and Lessen technique has been painstakingly intended to have minimal measure of time or space intricacy.

To further grasp the architecture of MapReduce, let's talk about its phases:
The Map phase and Reduce phase make up the bulk of the two phases that make up the MapReduce task.

- **Map:** As its name suggests, its basic function is to organize information data into key-value matches. This information can be a key-value pair with the location ID as the key and the value representing the data that the guide actually stores. Each key-value match of these pieces of information is subject to a Memory Vault execution of the Map() function, resulting in an intermediate key-value pair that serves as input to the Reduce or Reducer() function.
- **Reduce:** The middle key-value matches are arranged and rearranged prior to being shipped off the Reduce () capability as the input for Reducer. As indicated by the reducer calculation made by the designer, reducer aggregates or gatherings the data relying upon its key-value pair.

How the task tracker and job tracker handle MapReduce:

- **Job Tracker:** Since there may be many data centers available, Job Tracker's function is to manage all assets and jobs across the group and schedule each Job Tracker card that operates in that data center.
- **Task Tracker:** A task tracker can be thought of as a real slave that executes the instructions given by the job tracker. Each hub in the group that can do map-and-reduce work is equipped with this task tracker and deployed according to the job tracker guidelines.

The work history server is another important component of MapReduce engineering. The Job History Server is a daemon cycle that recovers and blocks verifiable data about jobs or applications. For example, the job history server stores logs delivered during or after job execution.

- **Why we need DFS**

You could wonder why we need a DFS when we can store files up to 30 terabytes in a single machine. This is so that systems' disk capacities can only rise so much. Processing enormous datasets on a single machine is inefficient, so if you manage to handle the data on a single system, you'll run into processing issues.

Let's use an example to better grasp this. Let's say you need to handle a file that is 40 TB in size. That will take 4 hours to process it all on a single computer, but what if you use a DFS? (Distributed File System). In that situation, as you can see in the graphic below, a cluster of 4 nodes is used to spread a 40TB file, with each node storing a 10TB file. We require DFS since, with all of these nodes operating concurrently, the fastest processing time is only 1 hour.

2.2. Local File System Processing:

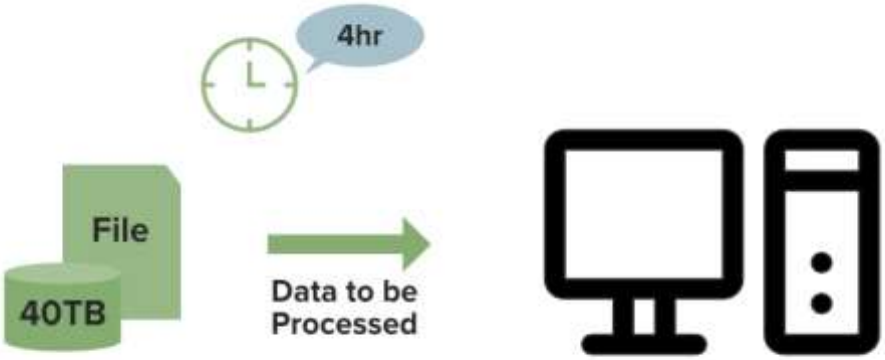


Figure 1 Local File Processing System

2.3. Distributed File System Processing:

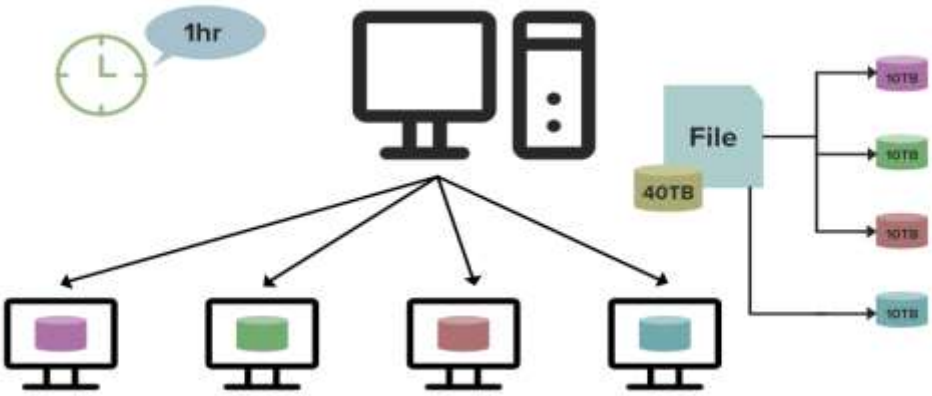


Figure 2 Distributed File Processing System

2.4. Overview – HDFS

Since you currently logical comprehend what a record framework is, we should begin utilizing HDFS. A Hadoop group utilizes HDFS (Hadoop Disseminated Record Framework) as its stockpiling stage. It is fundamentally expected for use with minimal expense gadgets known as "ware Equipment" that help conveyed document frameworks. How HDFS is built makes it more leaned to incline toward putting away data in large blocks rather than little ones. Hadoop's HDFS offers the capacity layer and different gadgets in the Hadoop bunch adaptation to non-critical failure and high accessibility.

Hadoop can work better and more reliably with a basic awareness of its parts, and HDFS' ability to handle large volumes, speeds, and diversity of large-scale data. Each data block in HDFS is 128 MB, but that can be changed to resolve the issue by modifying the hdfs-site.xml entry in the Hadoop catalog. HDFS stores data as blocks.

2.4.1. Important features of HDFS (Hadoop Distributed File System)

The open-source Hadoop Distributed File System (HDFS) is used to store and analyse huge datasets across a cluster of cheap hardware. The ease of access to the files kept in the system makes employing HDFS one of its key benefits. Moreover, HDFS offers high availability and fault tolerance, guaranteeing that the data will still be accessible even in the event of hardware issues or other disturbances. Another benefit is its scalability, which enables customers to scale up or down nodes in accordance with their needs. With the help of several datanodes, HDFS stores data in a distributed fashion. Moreover, this capability offers replication, removing any worry about data loss. Because HDFS is made to store enormous volumes of data—in the petabyte range—it is very dependable. Moreover, Name Node and Data Node include built-in servers that make it easier to get cluster information. Moreover, HDFS has high throughput, which is necessary for effectively processing big information. Because to these characteristics, HDFS is frequently chosen by many businesses to manage massive data.

2.4.2. HDFS Storage Daemon’s

As is well knowledge, Hadoop utilizes the master-slave design of the MapReduce algorithm, and HDFS features NameNode and DataNode components that follow a similar pattern.

1.NameNode(Master)

2. DataNode(Slave)

1. NameNode : In the Hadoop group, the NameNode acts as an expert and coordinates the Datanodes (slaves). A Namenode's basic role is to store metadata, or simply data about data. Exchange logs that record client actions within Hadoop bundles can serve as metadata.

In order to find the closest DataNode and respond more quickly, the Namenode stores data about DataNode ranges (block number, block id) as a function of metadata. DataNodes receive guidelines for activities such as Make, Repeat and Elimination from Namenodes.

While our NameNode serves as a Master, it has to be powerful enough to manage or direct all of the slaves in a Hadoop cluster. All of the slaves, or DataNodes, send heartbeat signals and block reports to Namenode.

3. **DataNode:** With Hadoop groups acting as slave DataNodes, DataNodes are primarily used for data capacity. Their numbers can range from 1 to 500 or more. The more DataNodes a Hadoop bundle contains, the more data it can store. Second, DataNodes are suggested to have high capacity limits to hold many document blocks. Following the NameNode guidelines, Datanodes perform exercises such as creation, deletion, etc.

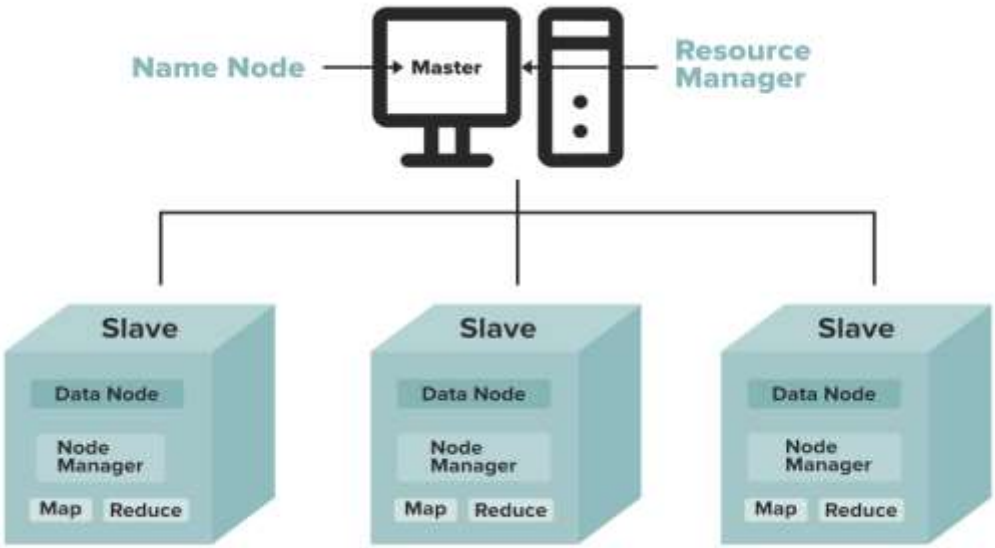


Figure 3 HDFS Storage Daemon's Model

3. **Dataset**

We gathered a dataset with a number of factors in order to analyze student performance. The participant's gender, which was divided into two groups as male and female, was the first variable we gathered. Race or ethnicity of participants divided into five groups (Group A, Group B, Group C, Group D, and Group E) was the second variable collected. The participant's parental education level was her third variable that we investigated, and fell into six categories.

Bachelor's degree, some colleges, master's degree, associate's degree, high school, some high school. The participant's lunch, which was divided into two categories—standard and free/reduced—was the fourth variable we gathered. The participant's participation in a test preparation course, which was divided into two groups: none and finished, made up the fifth variable we gathered. Lastly, we gathered information on the participant's arithmetic, reading, and writing scores. We intend to learn more about the elements that influence student success through the analysis of this dataset.

4. Methodology

Many factors present in the dataset for performance prediction are essential for projecting student performance with accuracy. Mappers are used to map the features, and reducers are given the results as input in order to analyze this data and extract the pertinent features. The machine learning algorithms are given a file containing these features by the reducers once they have extracted the appropriate features from the input.

We may carefully analyze and anticipate student performance using machine learning techniques based on the pertinent features that were retrieved. These algorithms make use of the attributes to identify trends and connections that may be used to predict how well the student will do in the future. By doing this, we may produce models that can accurately predict performance and acquire important insights into the variables that affect student success.

In the end, this strategy can assist institutions and educators in making defensible choices on the best ways to serve their students and enhance their academic performance. Accurately forecasting student performance allows educators to create interventions and teaching methods that are specific to each student's requirements, enhancing both academic success and general wellbeing. As a result, using machine learning algorithms to forecast student performance is a potent tool that can help students achieve successful educational results.

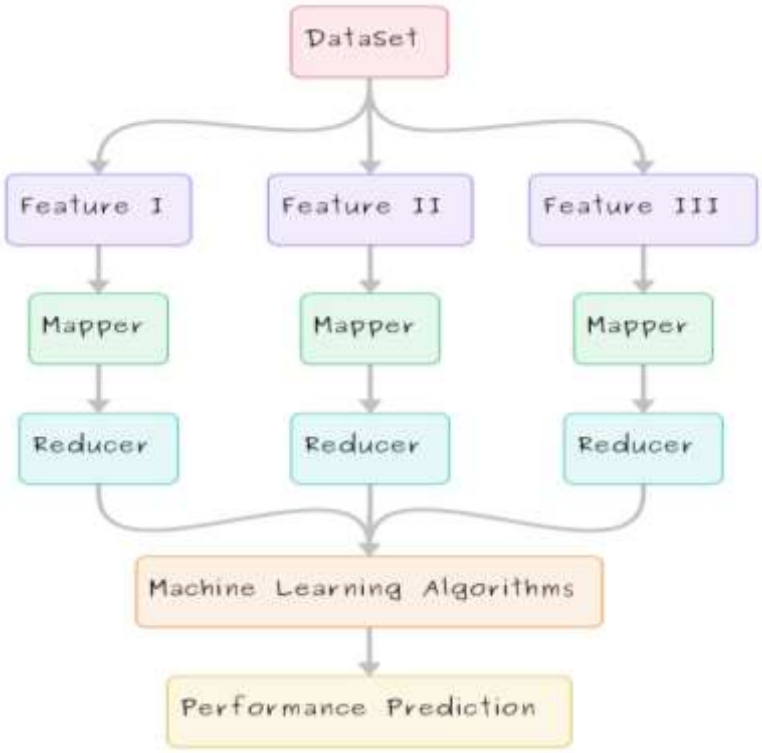


Figure 4 Machine Learning Model

The Mapper-Reducer function takes the data and extracts different aspects, such as the Gender Distribution, which shows the proportion of male to female students in a school.

4.1. Model Architecture

An essential part of the Hadoop ecosystem that offers dependable and scalable storage for huge data is the Hadoop Distributed File System (HDFS). With its many distinctive qualities, HDFS is a well-liked option for managing big information. The solution of the node failure issue in a Hadoop cluster is one of the primary objectives of HDFS. Node failure is a possibility in a cluster since there are several nodes made of commodity hardware in it. By including tools for detecting and recovering from node failures, HDFS addresses this problem. Maintaining data coherence and ensuring that computational processes are carried out effectively present a substantial additional barrier when working with huge datasets. Because of HDFS's ability to manage files ranging in size from gigabytes to petabytes, huge volumes of data may be processed and stored on a single cluster. HDFS lessens network congestion and boosts system throughput by carrying out computational activities close to where the data is stored.

Moreover, HDFS is portable, allowing it to run on several hardware and software systems. HDFS is now more responsive to shifting technological settings because to this capability. A straightforward coherency paradigm that enables the write-once-read-many access pattern for files is also implemented by HDFS. This methodology makes sure that just data is added to files; they are written and closed without being changed. The MapReduce framework is ideally adapted to cope with such a file format, and this assumption helps to reduce the problem with data coherency. Overall, HDFS's distinctive properties make it a crucial part for managing massive data and overcoming its difficulties. HDFS is a desirable alternative for

maintaining and processing huge data due to its capacity to withstand node failure, enormous datasets, and its straightforward coherency architecture.

5. Results

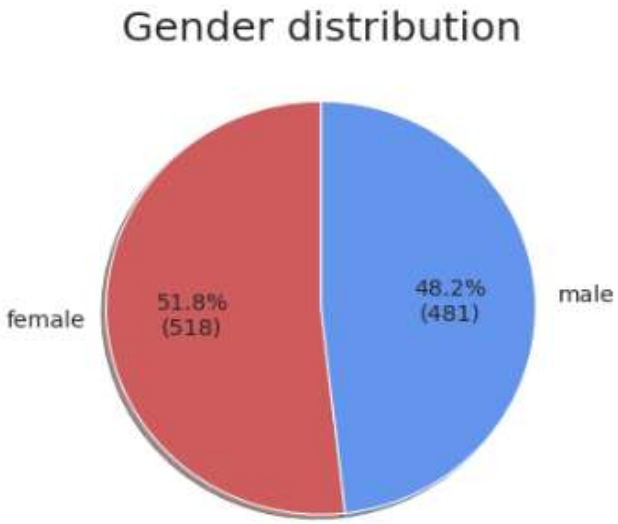


Figure 5 Gender Distribution

The graphic below is an example of how the extracted features were used in a multivariate analysis to highlight the correlations between the individual subject scores in Writing, Reading, and Math for each ethnic group. If there are any regular gaps or trends between these variables, this analysis can assist find them.

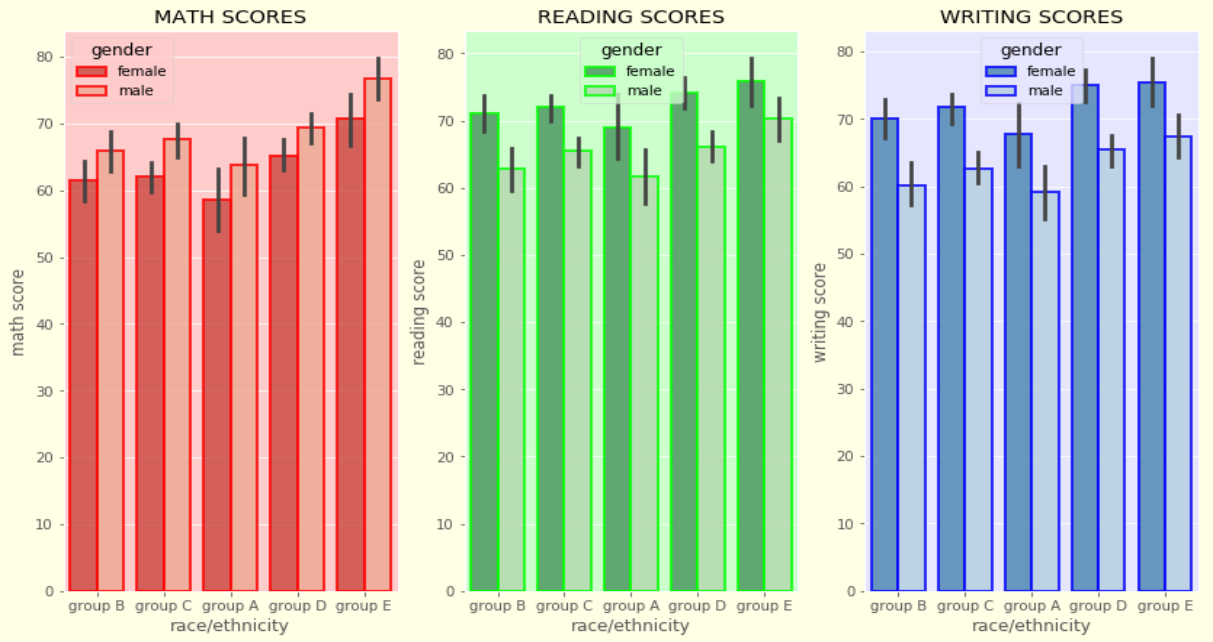


Figure 6 Extracted Features in a Multivariate Analysis

This picture demonstrated the connection between grades and the lunch recipients' readiness for the course.

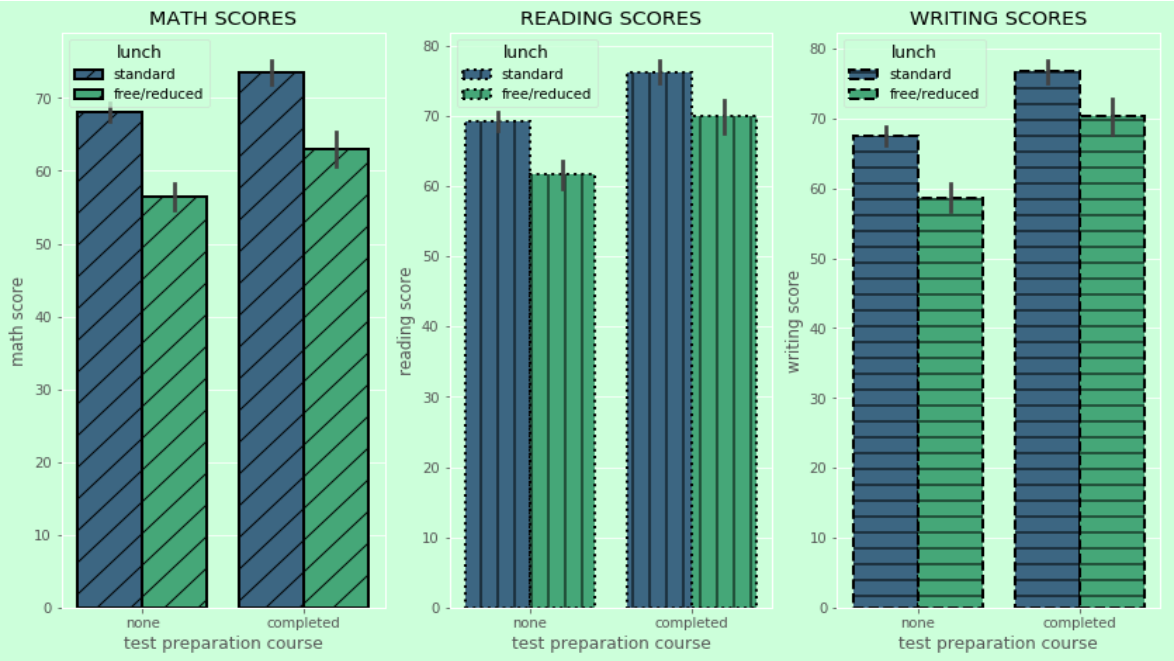


Figure 7 Connections between Grades and the Lunch Recipients' Readiness

5.1. Test preparation course

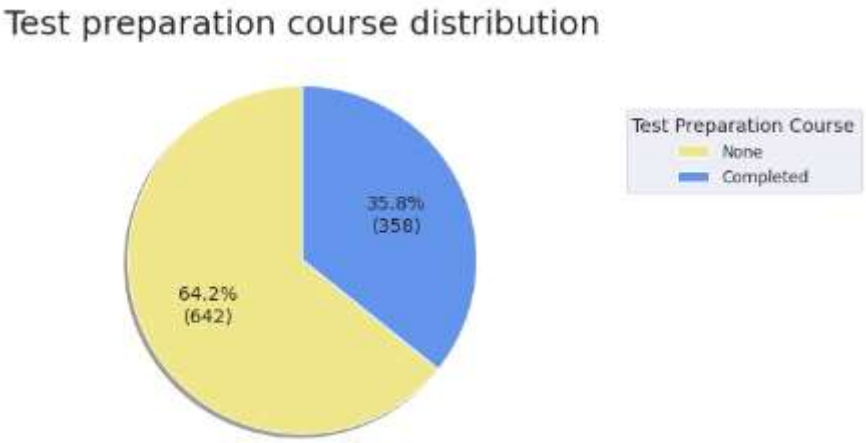


Figure 8 Test Preparation Course Distributions

Most students did not take the test preparation course.

5.2.Machine Learning Algorithms

These are machine learning models and their prediction performance

5.2.1. Linear Regression

One dependent variable and at least one independent variable are established using the direct regression fact method. The value of the dependent variable can be predicted using this

strategy by obtaining the values of at least one independent variable. Finding direct associations between dependent and independent variables is central to direct recurrence. The dependent variable is presented as the direct ability of the independent variable in the direct recurrence model. The straight ability is demonstrated in the situation y = mx + b where y is the dependent variable, x is the independent variable, m is the slope, and b is the catch. The limits determined from the data are slope and catch.

The main goal of direct regression is to find the best straight line to understand the relationship between dependent and independent variables. The line that limits the number of squared errors between the actual and expected values of the dependent variable is said to be the best fit.

5.2.2. Lasso

Least Absolute Shrinkage and Selection Operator, or Lasso, is a regularization and variable determination strategy for regression analysis. It is a linear regression model with a penalty term added to the cost function that forces some of the coefficients to be zero in order to promote sparse solutions. The two main applications of the Lasso regression method are feature selection and regularization. By decreasing the less significant variables towards zero, the Lasso regression approach in feature selection assists in identifying the most significant features or variables that are relevant to the result of interest. This can be especially helpful when there are many of predictor factors in high-dimensional datasets.

By reducing the regression coefficients to zero during regularization, the Lasso approach aids in lowering model complexity and avoiding overfitting. Overfitting is a condition when a model performs poorly when applied to fresh data because it is too complicated and fits the training data too well.

5.2.3. Ridge

A common technique for modeling connections between a response variable and a group of predictor variables is ridge regression. It is a linear regression model with a penalty term added to the cost function to reduce the regression coefficients toward zero and prevent overfitting. Similar to the Lasso technique, the Ridge regression approach employs the squared value of the regression coefficients rather than the absolute value. A tuning parameter,, which defines the trade-off between the model's fit and complexity, controls the severity of the penalty term.

Ridge regression can handle strongly correlated predictor variables, which is one of its main benefits. Highly correlated predictor variables in standard linear regression might result in unstable and incorrect estimations of the regression coefficients. Yet, the penalty term in Ridge regression aids in stabilizing the results by lowering their variance.

5.2.4. K-Neighbors Regressor

A well-liked non-parametric regression method for modeling relationships between a response variable and a group of predictor variables is K-Nearest Neighbors (KNN) regression. KNN regression does not presuppose a certain functional form for the link between the response and predictor variables, in contrast to parametric regression methods. Finding the k nearest neighbors to a particular test observation in the training data and utilizing the average or weighted average of those replies as the prediction for the test

observation is how the KNN regression technique operates. K is a hyperparameter whose value can either be fixed or determined by cross-validation. The fact that KNN regression can handle non-linear interactions between the response and predictor variables is one of its main benefits. This is so because KNN regression uses the data itself to estimate the relationship rather than assuming a certain functional form for the association.

5.2.5. Decision Tree

A well-liked non-parametric supervised learning technique for classification and regression applications is the decision tree. Each internal node in a choice tree addresses a test on a component, each branch addresses the experimental outcome's, and each leaf node stands for a class name or a mathematical value. The dataset is divided into smaller subsets based on the values of the characteristics, and the tree is constructed recursively by starting with the root node, which represents the complete dataset. Decision trees have the benefit of being simple to comprehend and interpret, which is one of its key advantages. Decision trees may be seen, and it is simple to deduce from their structure the rules that were used to categorize or forecast a new observation. Decision trees are thus a helpful tool for producing insights from data and for exploratory data analysis.

5.2.6. Random Forest Regressor

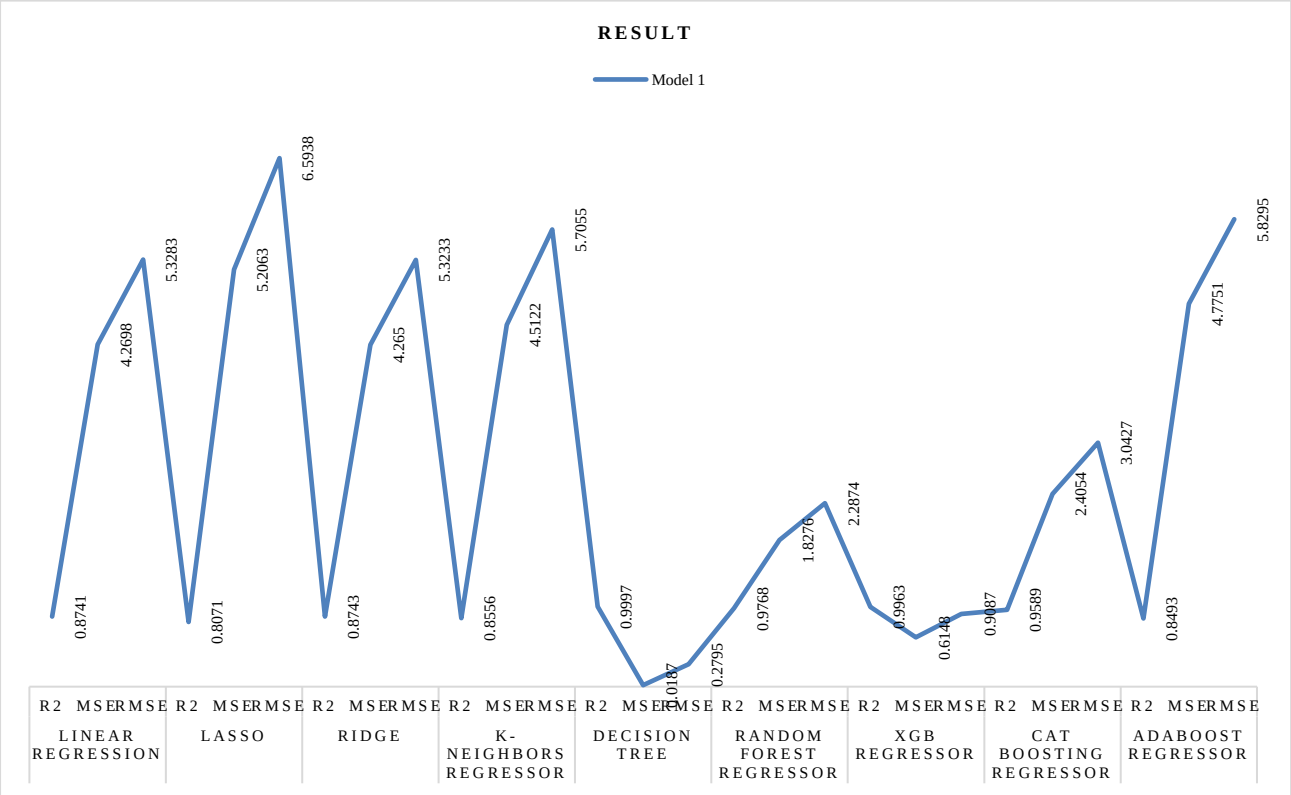
A well known group learning method for relapse applications is the Random Forest Regressor. It depends on the possibility of choice trees, yet builds various choice trees and consolidates their output to deliver forecasts instead of creating a solitary decision tree. Each tree in a Random Forest Regressor is constructed using a randomly chosen portion of the data, and a randomly selected subset of characteristics is taken into account at each split. This adds unpredictability to the model, which helps to lessen overfitting and enhance the model's generalization capabilities. Moreover, the Random Forest Regressor employs a method known as bagging, which entails randomly selecting data and replacing it with new data to produce several bootstrap samples. The final forecast is achieved by averaging the predictions of all the decision trees in the forest. Each bootstrap sample is used to train a single decision tree.

The Random Forest Regressor's reduced propensity for overfitting against a single decision tree is one of its key advantages. This is due to the fact that the random selection of features and data subsets minimizes correlation across the trees, aiding in the model's ability to generalize.

5.2.7. XGB Regressor

Extreme Gradient Boosting Regressor (XGB Regressor), a powerful machine learning technique, is widely used for regression applications. It is an evolution of gradient boosted regression that combines boosting and bagging approaches to create highly accurate predictive models. With the XGB Regressor, a model is created by integrating several weak models—typically decision trees—into one strong model. Decision trees are incrementally added to the model as part of the algorithm, and each new tree seeks to fix the flaws of the preceding trees.

The ability of XGB regressors to handle complex and nonlinear interactions between input features and target variables is one of its main advantages. In addition, you can handle data anomalies and missing values.



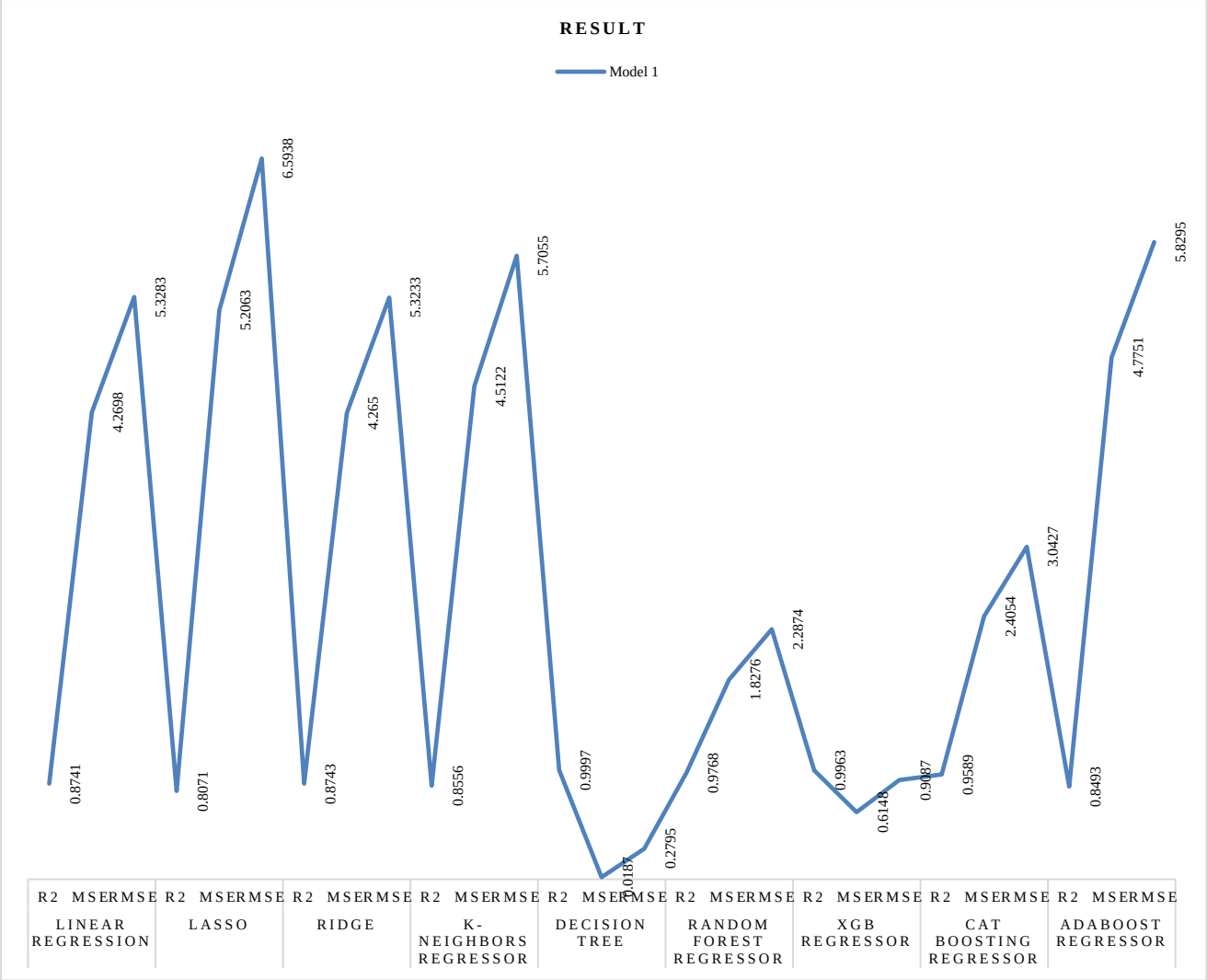


Table 1 Model Name and its R2 Score

Model Name	**R2_Score**
Ridge	0.880593
Linear Regression	0.879159
Random Forest Regressor	0.857040
CatBoosting Regressor	0.851632
AdaBoost Regressor	0.845403
Lasso	0.825320
XGB Regressor	0.821589
K-Neighbors Regressor	0.783193
Decision Tree	0.737855

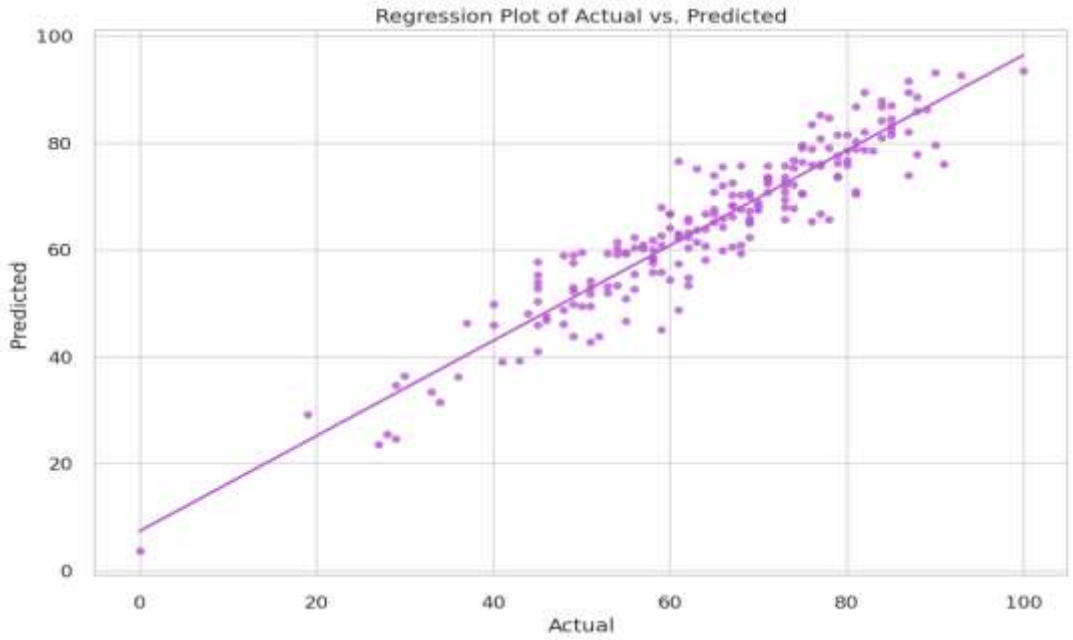


Figure 9 Regression Plot Image

6. Conclusion

This study investigates how different demographic characteristics affect pupils' academic achievement. The results show a gender divide, with men having a greater sex ratio than women. Female students often thrive in reading and writing, while male students typically perform better in arithmetic. Except for Group E, which scored better than other groups, there was no obvious influence on race or ethnicity. Also, the study demonstrates that students who choose the free/reduced lunch option typically received poorer grades than those who opted for the regular lunch option. The majority of students who started test preparation classes did not finish them, and those who did had higher grades than those who did not. It's interesting to note that test scores for youngsters did not significantly depend on parental education levels. These results show that in order to overcome the differences in academic achievement based on demographic characteristics, policymakers and educators need to adopt a more all-encompassing strategy.

The local networked system and the cloud-based system were chosen as the experimental platforms for idea validation and algorithm implementation. Prior to the algorithm's real implementation in the cloud-based system, the algorithm was tested on the local network system. The same Hadoop framework has been setup for both the local network system and the cloud-based system. The Hadoop framework was chosen because of the industrial and academic adoption of Hadoop. Using Python classes, the method has been run in the Hadoop framework. The python classes were mapped using three classes. Driver, Mapper, and Reducer are the names of these three classes. The primary class that initializes all of the input parameters is the Driver class. The mapping logic for the map stage of MapReduce is included in the Mapper class, a Python class. Both data slicing and the creation of key-value pairs are components of the mapping logic. The reducing logic for the MapReduce stage is included in the Reducer class, a Python class. The reducing logic comprises sorting and processing key-value pairs in accordance with the coding for the reducing logic. The data sets

from the US Census and Amazon's movie reviews were both used in this study. There are statistics from many different domains in the US census data collection. The business, economy, employment, government, health, housing, income, poverty, and population are the data domains. The reputable firm Amazon collected the data for the Amazon movie reviews data collection. Both data sets were chosen for this research because they were trustworthy and accessible in the public domain. Evaluation frameworks for determining algorithmic behaviour are provided in the next chapter.

7. Discussion

A technique for choosing the best solution has been implemented in assessment frameworks based on software/hardware variants. The determination of workload and workload restructuring were topics of additional study, which demonstrated how workload restructuring improved performance.

Performance benchmarks for MapReduce has been examined by Chen, Y., Ganapathi, A., Griffith, & Katz. They had created a language for characterizing MapReduce workloads following the examination of benchmarks. They identify a deficiency of two traces that are accessible to MapReduce researchers. They asserted that it would be impossible to examine several suggested MapReduce benchmarks without traces investigation. During their investigation of traces, they had employed two traces. Facebook and Yahoo are the sources of these traces. The 2009 Facebook trail covers 6 months and includes 1 million jobs. 30000 jobs are included in the three-week Yahoo trace. To enhance MapReduce's performance, the workload was reorganized. Unfortunately, it is not always viable to restructure the task. So, utilizing the methodology outlined in this study work, choosing the best configuration for software and hardware is a superior alternative.

Department cloud execution time was reduced, according to González-Vélez and Kontagora.

In order to establish if employing the departmental cloud would result in appreciable savings in execution time, they conducted an analysis of the performance of MapReduce. For their tests, they employed the Random Writer and sort algorithms. An method included in Hadoop known as Random Writer has been referred to be a data set generator. They had mostly used the sort algorithm in their experiments. They created a departmental cloud with similar computers, or nodes, and ran tests on various nodes in it. The departmental cloud outperforms the regular cloud, they found after comparing the two.

References

[1] Veiga, J., Expósito, R. R., Pardo, X. C., Taboada, G. L., &Tourifio, J. (2016, December). Performance evaluation of big data frameworks for large-scale data analytics. In 2016 IEEE International Conference on Big Data (Big Data) (pp. 424-431). IEEE.

[2] Buono, D., Danelutto, M., &Lametti, S. (2010). Map, reduce and mapreduce, the skeleton way. Procedia computer science, 1(1), 2095-2103.

[3] Chen, Y., Ganapathi, A., Griffith, R., & Katz, R. (2011, July). The case for evaluating mapreduce performance using workload suites. In 2011 IEEE 19th annual international

symposium on modelling, analysis, and simulation of computer and telecommunication systems (pp. 390-399). IEEE.

[4] González-Vélez, H., & Kontagora, M. (2011). Performance evaluation of MapReduce using full virtualisation on a departmental cloud. International Journal of Applied Mathematics and Computer Science, 21(2), 275-284.

[5] Veiga, J., Expósito, R. R., Taboada, G. L., &Touriño, J. (2016). Analysis and evaluation of MapReduce solutions on an HPC cluster. Computers & Electrical Engineering, 50, 200-216.

[6] Hilbert M, López P (April 2011). "The world's technological capacity to store, communicate, and compute information" Science. 332 (6025): 60–5.

[7] Mikalef, P., Krogstie, J., Pappas, I. O., &Pavlou, P. (2020). Exploring the relationship between big data analytics capability and competitive performance: The mediating roles of dynamic and operational capabilities. Information & Management, 57(2), 103169.

[8] "Big Data facts – How much data is out there?"www.nodegraph.se/big-data- facts. Retrieved 19 February 2020

[9] Viles, E., Ormazábal, M., &Lleó, A. (Eds.). (2018). Closing the Gap Between Practice and Research in Industrial Engineering. Springer International Publishing.

[10] Khan, M. A., Cherif, W., Filali, F., &Hamila, R. (2017). Wi-Fi Direct Research-Current Status and Future Perspectives. Journal of Network and Computer Applications, 93, 245-258.

[11] Walnycky, D., Baggili, I., Marrington, A., Moore, J., &Breitinger, F. (2015). Network and device forensic analysis of android social-messaging applications. Digital Investigation, 14, S77-S84.

[12] Kott, A., Wang, C., &Erbacher, R. F. (Eds.). (2015). Cyber defense and situational awareness (Vol. 62). Springer.

[13] Mansfield-Devine, S. (2016). Ransomware: taking businesses hostage. Network Security, 2016(10), 8-17.

[14] Ranjan, J. (2009). Business intelligence: Concepts, components, techniques and benefits. Journal of Theoretical and Applied Information Technology, 9(1), 60-70.

[15] Das, T. K., & Kumar, P. M. (2013). Big data analytics: A framework for unstructured data analysis. International Journal of Engineering Science & Technology, 5(1), 153.

[16] Welcome to Apache™ Hadoop®! (n.d.). Retrieved 19 February 2020, from https://hadoop.apache.org/

[17] Apache Hadoop. (n.d.). Retrieved 19 February 2020, from http://www.cloudera.com/content/cloudera/en/about/hadoop-and-big-data.html

[18] Ghemawat, S., Gobioff, H., & Leung, S. T. (2003, October). The Google file system. In Proceedings of the nineteenth ACM symposium on Operating systems principles (pp. 29-43).

[19] Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified data processing on large clusters.

[20] Chitresh Verma and Dr. Rajiv Pandey. "Big Data Representation for Grade Analysis Through Hadoop Framework." Proc. of Confluence-2016 - Cloud System and Big Data Engineering. ISBN: 978-1-4673-8202-1