BIG DATA: STORAGE, ANALYSIS, AND IMPLEMENTATION

Melissa Pula and Omar A-Azzam

Computer Science and Information Technology Department (CSIT), Saint Cloud State University (SCSU), Saint Cloud, MN, USA

ABSTRACT

We are in an age where data is not only readily available, but abundant. However, it is what we do with this data, through analyzing it, that is most important. While traditional means of data analytics works well for smaller amounts of information, it may not be able to handle larger amounts, called big data. In this paper, we will discuss the different ways in which big data can best be stored, what methods work best in analyzing it, and how the results of this analysis can be implemented in real-world scenarios.

KEYWORDS

Big Data, Analytics, Machine Learning, Data Mining

1. INTRODUCTION

Information obtained from data can be incredibly useful and important to many professionals. From a business perspective, the insights that can be gained from analyzing such data are a great asset and of great value to the business's future outlooks. It enables them to accurately predict what products consumers will buy and how much, minimizing waste and maximizing sales profits.

There are really two topics to consider when it comes to Big Data Analytics: Big Data, and Analytics. Big data is comprised of the three V's – variety, volume, and velocity. What this entails is that big data can be defined as an extremely large volume of data and datasets coming from multiple sources at great speed. This data can be either structured or unstructured and is so voluminous that it cannot be processed by traditional data processing software. Analytics is comprised of four main types: descriptive, predictive, diagnostic, and prescriptive. A description of each type, as well as how they can be implemented, will be included in this paper.

The purpose of this paper is to research big data analytics with the goal of obtaining a complete understanding of what exactly comprises big data analytics, as well as how big data and analytics are used in conjunction with each other. To do so, we will run various analytical programs on data using traditional data processing, as well as one program on a "big data" dataset, and then compare the preprocessing needed for each, as well as the time it took for the process to complete.

David C. Wyld et al. (Eds): SEAU, NLMLT, ITCS, CST, DMS -2023
pp. 29-41, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.132003

2. LITERATURE REVIEW

2.1. Storage

In this section, we will discuss big data storage technologies and their capabilities, as well as areas in which these technologies may be lacking. The following table illustrates the position of big data storage within the Big Data Value Chain.

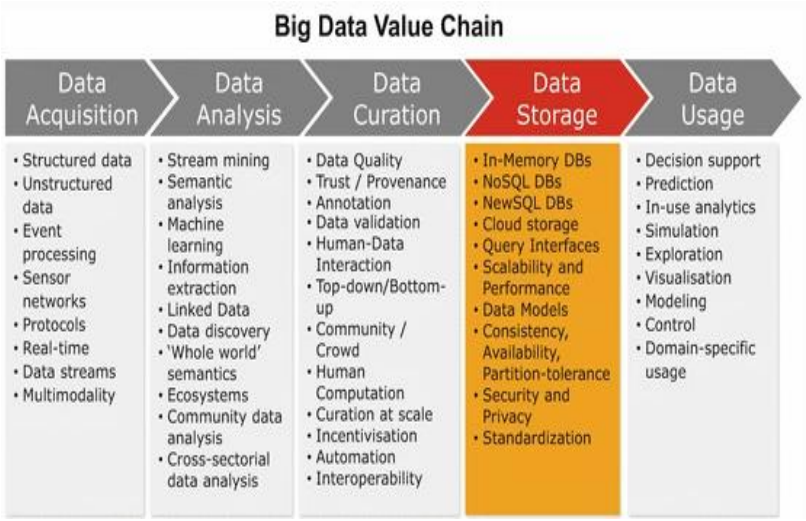


Figure 1. Big Data Value Chain [1]

Big data storage technologies address the three V's - variety, volume, and velocity. They do not, however, fall into the category of relational database systems. This isn't to say that relational database systems do not also address the three V's, but rather alternative storage technologies are often less expensive and more efficient options.

2.1.1. Challenges

The challenge with variety in big data storage relates to the level of effort required to work with and integrate data that is being pulled from a large number of resources. In contrast, the challenge with velocity refers to query latencies, which is of particular importance when there are high rates of incoming data. And thirdly, the challenge with volume is in reference to the sheer mass of data being utilized. This is addressed by making use of a distributed architecture.

Some technologies that provide advances in big data performance, usability, and scalability, are those that use Hadoop, such as MapR, Cloudera, and various NoSQL database vendors. These technologies enable virtually boundless volumes of data to be stored at a lower cost and with lower operational complexity.

Despite the advances in such data storage technologies, there still remain untapped potential and limitations. One such untapped potential is that storage of big data and the technologies used are key enablers in advanced analytics and can be used to change society and the way in which business decisions are made. Of particular interest would be in non-IT sectors. These sectors, such as energy, for example, lack experts in big data. Storage technologies could potentially enable new value-generating analytics to them.

One area in which big data storage technologies are lacking is in privacy and security. While there are several solutions in place, as well as a number of projects addressing this issue, individual protection and security is lagging. To rectify this issue, considerable research to better understand the room for data misuse, and how it can be prevented, is required.

2.1.2. Technologies

The more big data technologies emerge into different sectors of society, the more data-driven society is becoming. Because of this, it is important that data, and big data in particular, is stored in a way such that it is cost-efficient, easily accessible, and can be analyzed and implemented to better help make decisions.

Some of the current state-of-the-art data storage technologies are NoSQL Databases, Distributed File Systems, Big Data Querying Platforms, and NewSQL Databases. Hadoop File System is an example of a distributed file system. Using these systems offers the capability to store vast amounts of data that is unstructured. It is designed for large data files and is well suited for taking in data quickly and then processing it in bulk.

NoSQL Databases are probably the most important family when it comes to data storage technologies. They are designed for scalability and, when compared to relational databases, often use non-standardized, low-level query interfaces. This makes them more difficult to integrate when existing applications expect an interface to be SQL. They also lack standard interfaces, which makes it harder to switch vendors.

The distinguishable data models NoSQL databases use are key-value stores, columnar stores, document databases, and graph databases. With key-value stores, data is allowed to be stored in a schema-less way. Data objects can be structured or unstructured and, in fact, don't even need to share the same structure since no schema is used. Data objects are also accessed as a single key.

Columnar stores are when database management systems store data tables in sections of columns, rather than rows. The data is then indexed by a column key, row key, and time stamp, and the value is represented by a string type. The data is then accessed by column families.

Unlike values in key-value stores, document databases are structured. There is, however, not a common schema requirement like there is in a relational database. Because of this, document databases are referred to as semi-structured. As with key-value stores, queries can be made to documents using a unique key, but in addition, documents can also be queried by their internal structure.

As the name implies, graph databases store data in graph structures. This makes them more suitable for storing data such as network graphs, which are highly associative.

While they differ in performance and approach, when accessing data, big data querying platforms typically offer an SQL-like interface. An example would be Hive, which allows querying to structured files through an SQL-like query language. However, it executes these queries by translating them in MapReduce jobs, which results in a high latency for not only large datasets, but smaller sized ones as well. The benefit, however, is that the SQL-like interface and flexibility of Hive easily evolve its schemas.

In contrast to Hive's high latency queries, Impala, which is another technology that utilizes an SQL-like interface, is designed to execute queries with low latencies. While it re-uses the same metadata as Hive, it uses its own distributed query engine to achieve these low latency queries.

NewSQL Databases aim for a scalability that is comparable to NoSQL databases, but in a more modern form of a relational database. It does this while also maintaining traditional database system transactional guarantees. In a NewSQL database, SQL is the primary mechanism when it comes to application interaction. They also contain atomicity, consistency, isolation, and durability (ACID) support for transactions and a mechanism to control non-locking concurrency. In addition, NewSQL databases contain an architecture that is capable of running a large number of nodes at a much higher performance level and without suffering bottlenecks.

An additional option for big data storage is cloud storage. There are many big-name companies, such as Google and Amazon, that have developed their own cloud platforms. Other companies, such as HP and Dell, use an open source platform called OpenStack to build their cloud systems on.

Cloud storage is beneficial for both end users and enterprises. An end user can access data, such as documents, from anywhere when those documents have been stored on a cloud platform. It can also be utilized as a backup to data stored on a desktop. For enterprises, cloud storage can offer better support, more flexible storage, and cheaper pricing.

One of the main concerns with cloud storage, however, is security. It is predicted that the majority of data will soon be stored on some cloud platform. While this is convenient for users and enterprises alike, it could also pose a severe security risk if important data, such as passwords, are compromised. It is much easier for a hacker to gain access to information in a cloud, since it is all online, than it is for them to break into a personal laptop. [2], [3], [4]

2.2. Analysis

The process of knowledge discovery in databases can be summarized by the following operations: selection, preprocessing, transformation, data mining, and interpretation/evaluation. This is illustrated with the below figure.

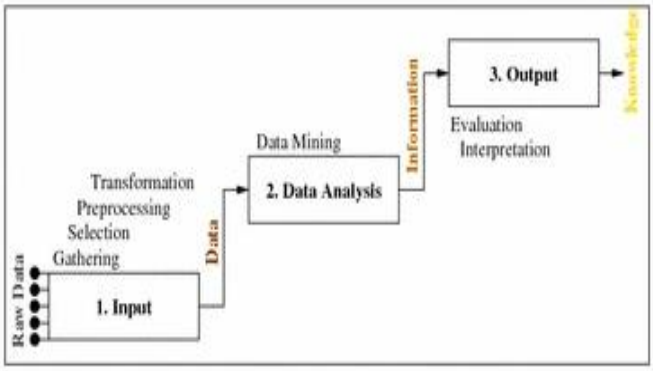


Figure 2. Knowledge Discovery Process [3]

Data input is where the gathering, selection, preprocessing, and transformation operators exist. In this area of knowledge discovery, the selection operator decides and selects what kind of data from the gathered information is required for analysis. In contrast, the role of the preprocessing operator is to clean and filter out the unnecessary or useless information from the dataset. Once this has been accomplished, the transformation operator then transforms the data into a data-mining-capable format.

The data analysis portion of knowledge discovery is then where the data mining takes place. The goal in this analysis is to find patterns or hidden information that can be gleaned from the data. It does so through use of algorithms designed specifically for the mining of data, such as the following:

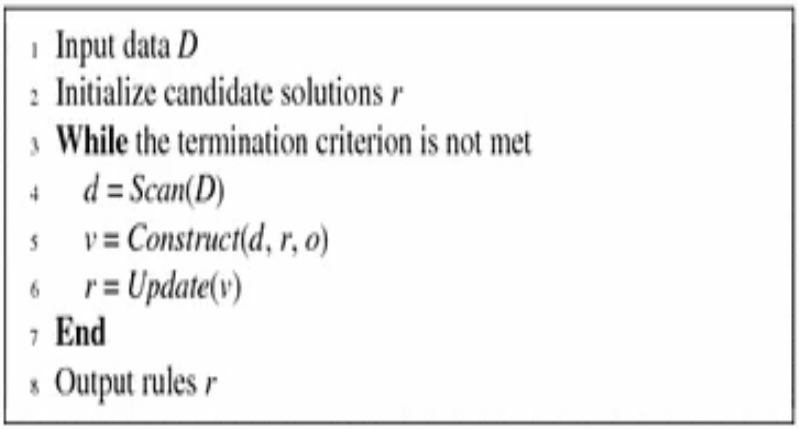


Figure 3. Data Mining Algorithm [3]

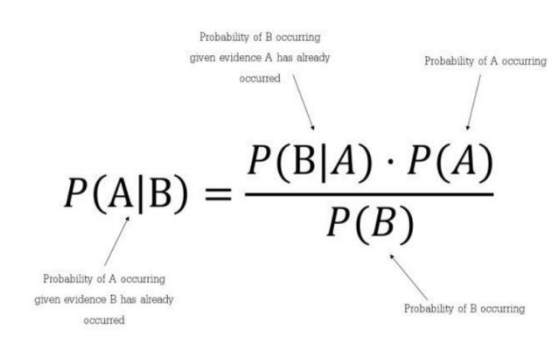
This algorithm illustrates what most data mining algorithms all contain: data input, initialization, data scan, and output. The scan, construct, and update operators iterate through until a certain termination criterion is met.

A very well-known method in data mining is called clustering. The result of this process is to separate an unlabeled dataset into k different groups. One such algorithm that accomplishes this is k-means [5]. An example of this algorithm is as follows:

Table 1. K-Means Algorithm [6]

Algorithm 1 *k*-means algorithm
1: Specify the number k of clusters to assign.
2: Randomly initialize k centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.

Another method in data mining is called classification. As opposed to clustering, classification relies on a labeled dataset and then constructs classifiers from it. These classifiers are then used to classify any unlabeled into their like groups. An example algorithm of classification is the Naive Bayesian method. The theorem is illustrated as follows:



Equation 1. Naïve Bayes Algorithm [7]

Unlike both clustering and classification, the goal of sequential patterns and association rules is to find relationships between data. An example of this method is the Apriori algorithm. It could be used for something such as helping customers buy products. It consists of support, confidence, and lift. The support refers to the popularity of a certain product. In other words, how many times a product, let's say diapers, was purchased in relation to the total number of purchases. The confidence level refers to the probability that a customer who buys a product, such as diapers, will buy another product as well, such as milk. So, it would be the number of times both diapers and milk were purchased in comparison to how many times diapers were purchased. And finally, lift refers to the increase in ratio of milk when diapers are also offered.

Two vital operators of the output are interpretation and evaluation. The results should be measured and then accurately and visually displayed in a way that is easy to interpret. One way of doing this would be through a graphical user interface.

These methods of data analysis are typically sufficient for traditional data analytics, however, when it comes to big data, the approach and execution may need to be adjusted to accommodate the sheer mass of data being ingested. While big data means more information, it may not always mean that the information is useful. It may actually contain more abnormal or ambiguous data which may degrade the accuracy of the mining results. Because of these limitations, several issues arise with big data analytics, such as security and data quality.

When comparing big data to traditional data analytics, the volume of data will be largely increased, there will be a large quantity of data being pulled in a short amount of time in relation to velocity, and the variety of the data may greatly differ as well, all posing a challenge to big data analytics [2]. The following diagram illustrates a comparison between traditional data analysis and big data analysis on a wireless sensor network:

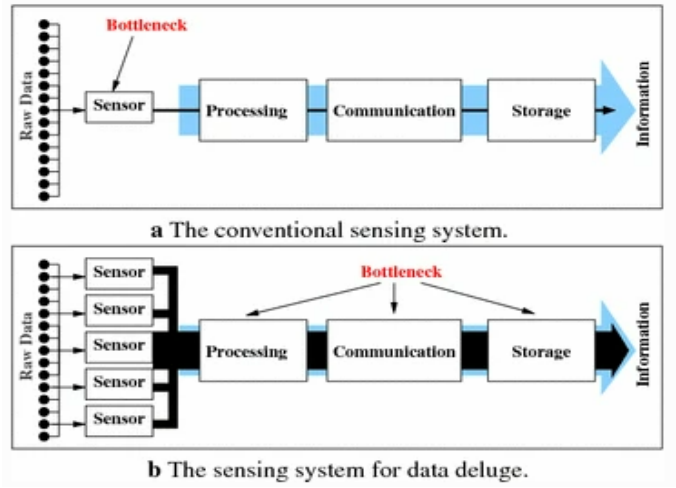


Figure4. Traditional vs Big Data Network Analysis [3]

2.3. Implementation

The output of data analytics can serve many purposes. Businesses can gain a great insight from data gathered on their customers or on the products and services people are looking for. It can help them make decisions on future products as well as help them understand current trends by analyzing past data. To help them accomplish this, there are four main types of data analytics: predictive, prescriptive, diagnostic, and descriptive. [8]

Predictive analytics is thought to be the most commonly used method in data analysis. Businesses, especially, can use predictive software to help them determine trends and see what products and services are currently being sought. [9], [10]

Similarly to predictive analytics, prescriptive analytics predict what products and services will be sought after in the future. It then enables businesses to make informed decisions on what they will be offering in the future and what they will decide to stop offering. It could also be used to help individuals make educated and informed decisions on stock purchases, property purchases, etc. [10]

Analyzing data from the past is called diagnostic data analytics. This can be an important method of data analysis in that it can help guide a business. Past information can be used to determine cause and event. It helps answer the question as to why something occurred. [10]

The best way to understand information and the analysis that has been done on it is through descriptive analytics. This type of analysis focuses on questions such as who, what, when, where, why, and what. It is the foundation of things such as dashboards and business intelligence tools, which would not exist without it. [10]

3. METHODOLOGY

To better understand the various ways in which data can be analyzed, we explored and created various coding files in which we ran data analysis algorithms. The methods we used are Naive Bayes, K-Means, and DBSCAN. In addition to traditional data analysis, we also experimented with big data analysis through use of a database consisting of images taken from drones. What makes this dataset “Big Data” is that it is approximately 3.5TB in size.

4. EXPERIMENTAL RESULTS

To learn more on the Naive Bayes theorem and how it works, we studied and wrote a program in Python. Our imports were as follows:

```
import numpy
import pandas
from math import sqrt, pi, exp
import matplotlib.pyplot as plt
from sklearn import metrics
```

Figure 5. Naïve Bayes Imports

We imported NumPy to gain use of its array computation capabilities. Pandas was imported to take advantage of its DataFrame, which provides two-dimensional, size-mutable, potentially heterogeneous tabular data. For various calculations throughout the program, Math was imported. To plot data and view results, Matplotlib was imported. And finally, for error prediction, we imported metrics from sklearn.

Next, we loaded a training and testing file into our program and then shaped the training file.Our third step was to start defining methods to help us calculate mean, standard deviation, probability, and utilize the gaussian algorithm. We then loaded and ran the testing file. To visualize the results, we constructed a confusion matrix. Next, wecalculated precision and recall,and then finally, we added code to save everything to a text document so that we could view the results.

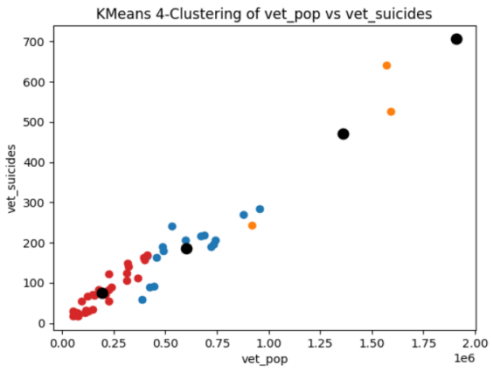
To understand the k-means method, we found a csv file containing statistics on veteran suicide by state. This contains information regarding how many veterans commit suicide each year in each individual state, how many total suicides per state there are, what age range the veteran is, etc. We used information from the year 2011 to conduct our experiment. Our imports were very similar to those that we imported for the Naive Bayes program:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans
```

Figure 6. K-Means Imports

Next, we loaded the csv file. The dataset contained null entries in various sections, which initially resulted in our program crashing. To avoid this, we replaced all null (NaN) values with zeros. We then extracted the data by column and then used NumPy to create an array with it. For this experiment, we only wanted to use veteran population, overall population, veteran suicides, overall suicides, and male veteran statistics.To help decide what size k should be for clustering purposes, wenext implemented the elbow method.The results toldus that 4 clusters would be the optimal choice because there isn't much change thereafter.

Now that we had that figured out, we wrote the code that would compute and display the actual clustering of our data. We ran it with k=4 and k=5 for comparison, and plotted veteran suicides and total number of veterans:



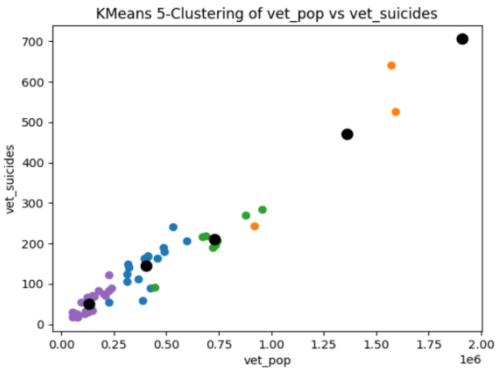


Figure 7. Results When K Equals 4 Figure 8. Results When K Equals 5

To experiment with DBSAN, we ran it on a simple array to show how it can display noise, or outliers (shown in blue data points), of a dataset [11], [12]:

```
from sklearn.cluster import DBSCAN
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

X = np.array([[5, 8], [6, 7], [6, 5], [2, 4], [3, 4], [5, 4], [7, 4], [9, 4], [3, 3], [8, 2], [7, 5]])

dbscan = DBSCAN(eps=2, min_samples=4).fit(X)

print(dbscan.labels_)

set(dbscan.labels_)
p = sns.scatterplot(data=X, x=X[:, 0], y=X[:, 1], hue=dbscan.labels_, legend="full", palette="deep")
plt.show()
```

Figure 9. DBSCAN

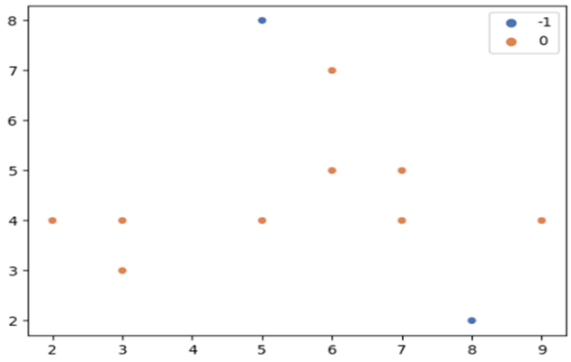


Figure 10. DBSCAN Outliers

All of these previously expressed experiments took a matter of seconds to complete and display the results. There was very minimal preprocessing involved, and the algorithms were fairly standard to understand, create, and implement. When testing these methods on big data, however, there was quite a bit more preprocessing and configuration that went into it. We used a

dataset called Functional Map of the World. It is comprised of approximately 3.5TB of drone aerial images [13]. The size of this dataset is what makes it "big data." Since it is an unlabeled dataset, we used clustering by means of the k-means, with the goal of sorting the dataset into groups of like images.

Some of the preprocessing that was necessary before analysis could be made was to resize each image to be the same size, convert each image from RGB to BGR, and convert to float32 format. We also applied PCA and t-SNE for visualization, and the elbow method to pick out the best value of k. Below is a list of our imports:

```
# for loading/processing the images
from keras.utils import load_img
from keras.applications.vgg16 import preprocess_input

# models
from keras.applications.vgg16 import VGG16
from keras.models import Model

# clustering and dimension reduction
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# for everything else
import os
import numpy as np
import matplotlib.pyplot as plt
from random import randint
from PIL import Image
import pandas as pd
import pickle
from progressbar import ProgressBar
```

Figure 11. Big Data Analysis Imports

We used Keras for image classification, sklearn for machine learning, OS to aid in navigation, NumPy for math, Matplotlib for plotting, PIL for imaging, Pandas for data analysis and manipulation, ProgressBar to implement a progress bar, and Seaborn for data visualization.

Our first step was to create an empty array and then iterate through the files and append each image found to the images array.Next, we loaded the VGG16 model.Our third step was to extract the image features.To ensure the images were extracted correctly, we then read in an image and displayed it.

After ensuring our images extracted correctly, we then set the image features to an array, printed the shape of the array, reshaped the array, and then printed the new array. Next, we obtained a list of all the file names and set those to an array, as well as the features, and reshaped them.We then reduced the number of dimensions in the feature vector and printed the before and after to show the difference.

Next, we created a DataFrame and used PCA Visualization.We then used the elbow method to see which value for k would work best.We also imported yellowbrick for elbow silhouette visualization.We then ran our k-means clustering algorithm. The results were as follows:

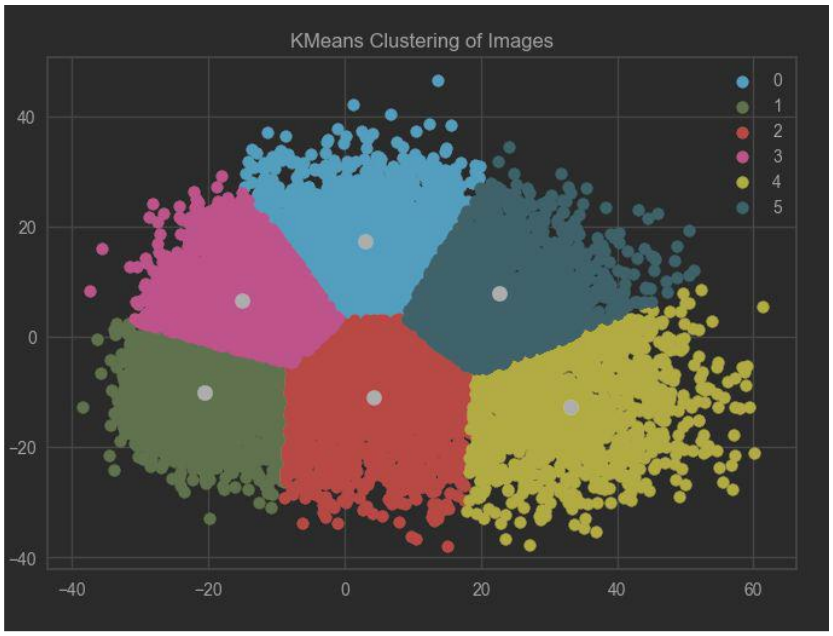


Figure 12. K-Means on Big Data Results

Next, we used silhouette_score to visualize the silhouette score.We then set an id and image to each group.And then implemented code to view each cluster. The following shows the first 30 images saved to cluster 1.

Figure 13. View Cluster 1

The main differences between traditional data analysis vs big data analysis were the data preprocessing and the amount of time it took to run each program. With traditional analysis, the size of the datasets were fairly small, there was minimal preprocessing, the code was fairly easy to implement, and it was near instant to process and run the data through the code. For the big data experiment, however, the dataset was 3.5TB in size, there was quite a bit of preprocessing involved, the code was much more difficult to create and implement, and it took days to convert and run through the algorithms.

5. DISCUSSION

The research we have conducted on big data analytics, such as the storage of big data, the analysis of it, and the implementation of that analysis through various modeling approaches, has

been extremely informative and helpful in gaining an overall understanding of big data analytics. We were able to experience first-hand, through experimentation, how to analyze data and display the results. In addition, we were able to experience the difference in analyzing traditional data vs big data through various exercises in analyzing and visualizing the results of data through the k-means, k-medoids, and DBSCANmethods.

6. CONCLUSION

Data analysis is an incredibly powerful and useful implementation for industries to incorporate. It can be implemented to show why something happened by analyzing data collected in relation to past information; it can help show current trends by collecting and analyzing data from current datasets; and it can help predict future outcomes and help businesses make informed decisions. Traditional data analysis can be done easily and within a shorter time span, whereas big data analysis takes a substantially longer amount of time to preprocess and run. Regardless, it is well worth the effort to enable businesses to take advantage of everything the results of this analysis has to offer.

REFERENCES

[1] A. Freitas and E. Curry, "Big Data Curation," https://www.researchgate.net/publication/280625426_Big_Data_Curation

[2] N. Elgendy and A. Elragal, "(PDF) Big Data Analytics: A Literature Review Paper," *ResearchGate*, Aug. 2014. https://www.researchgate.net/publication/264555968_Big_Data_Analytics_A_Literature_Review_Paper

[3] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *Journal of Big Data*, vol. 2, no. 1, Oct. 2015, doi: 10.1186/s40537-015-0030-3.

[4] M. Strohbach, J. Daubert, H. Ravkin, and M. Lischka, "Big Data Storage," *New Horizons for a Data-Driven Economy*, pp. 119–141, 2016, doi: 10.1007/978-3-319-21569-3_7.

[5] P. Arora, Deepali, and S. Varshney, "Analysis of K-Means and K-Medoids Algorithm For Big Data," *Procedia Computer Science*, vol. 78, pp. 507–512, 2016, doi: 10.1016/j.procs.2016.02.095.

[6] K. Arvai, "K-Means Clustering in Python: A Practical Guide," https://realpython.com/k-means-clustering-python/

[7] N. S. Chauhan, "Naïve Bayes Algorithm: Everything You Need to Know," https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html

[8] H. Watson, "Communications of the Association for Information Systems Tutorial: Big Data Analytics: Concepts, Technologies, and Applications," *Communications of the Association for Information Systems*, vol. 34, p. 65, 2014, doi: 10.17705/1CAIS.03462.

[9] J. Zakir, T. Seymour, and K. Berg, "BIG DATA ANALYTICS," *Issues in Information Systems*, vol. 16, no. 2, pp. 81–90, 2015, [Online]. Available: https://iacis.org/iis/2015/2_iis_2015_81-90.pdf

[10] "Can Data Analytics Improve Business Decisions?," *Oracle.com*, 2020. https://www.oracle.com/business-analytics/data-analytics/

[11] "Getting More Information About a Clustering — hdbscan 0.8.1 documentation," *hdbscan.readthedocs.io*. https://hdbscan.readthedocs.io/en/latest/advanced_hdbscan.html

[12] S. Yıldırım, "DBSCAN Clustering — Explained," *Medium*, Apr. 22, 2020. https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556

[13] Christie, Gordon and Fendley, Neil and Wilson, James and Mukherjee, Ryan, "Functional Map of the World", 2018, https://github.com/fMoW/dataset

[14] M. Q. Shabbir and S. B. W. Gardezi, "Application of big data analytics and organizational performance: the mediating role of knowledge management practices," *Journal of Big Data*, vol. 7, no. 1, Jul. 2020, doi: 10.1186/s40537-020-00317-6.

[15] L. Gordon, "Council Post: The Data Analytics Implementation Journey In Business And Finance," *Forbes*, Oct. 14, 2022. https://www.forbes.com/sites/forbestechcouncil/2022/10/14/the-data-analytics-implementation-journey-in-business-and-finance/?sh=6d42f3831828

AUTHORS

Dr. Omar Al-Azzam is an Associate Professor of Software Engineering in the Department of Computer Science and Information Technology (CSIT) at Saint Cloud State University (SCSU). Dr. Al-Azzam earned his BSc and MSc from Yarmouk University, Jordan and PhD from North Dakota State University (NDSU). Dr. Al-Azzam's main research interests are big data analytics, bioinformatics, and data mining.

Melissa Pula is a recent graduate from the Professional Science Master of Software Engineering (PSMSE) program at Saint Cloud State University (SCSU) in the Department of Computer Science and Information (CSIT) and works at TransImpact as a Web Developer. Mrs. Pula earned a BSc in Criminal Justice, a BSc in Mathematics with a Minor in Computer Science from Bemidji State University (BSU), and a PSM in Software Engineering from Saint Cloud State University (SCSU).