ETHICS OF TWEET CLASSIFICATION AND CLUSTERING UISNG UNSUPERVISED AND SUPERVISED LEARNING MODELS WITH ASSESSMENT METRICS

ZHEN XIN TAN RUAN

ETHICS OF MACHINE LEARNING

SYRACUSE UNIVERISTY
ENGINEERING COMPUTER SCIENCE

THURDAY MAY 13, 2021

SPRING 2021

ETHICS OF TWEET CLASSIFICATION AND CLUSTERING USING UNSUPERVISED AND SUPERVISED LEARNING MODELS WITH ASSESSMENT METRICS

Zhen Xin Tan Ruan

ABSTRACT

Clustering algorithms aim to group data points with similar properties, characteristics, or features in the same group. Thus, data points in different groups should have highly dissimilar properties or attributes. We can use clustering analysis to gain some valuable insights from the dataset by seeing what groups the data point fall into when we apply a clustering algorithm [5]. In tweet classification and clustering, tweets can be classified into different clusters based on their content and keywords. To classify them, researchers and developers have been using standard machine learning clustering techniques such as K-means and means shift. Like any other technology, this type of machine learning technique has positive and negative effects on our lives and society. In addition to the negative impacts, as data becomes easier to collect and as computer algorithms become more sophisticated, more advanced clustering techniques using machine learning will be developed [1] [2]. This sophistication of algorithms will raise various ethical concerns and questions about the application and use of this type of technology and dataset.

In this paper the current ethical challenges of clustering algorithms will be explored. The performance of several clustering algorithms using different assessment metrics will be studied and measured. This paper will provide a detailed explanation about the ethical concerns around clustering algorithms and datasets, along with an empirical clustering quality and runtime analysis of these learning methods on the dataset. To achieve this, we use the popular online platform Kaggle, where we downloaded the dataset from sentiment tweets. This includes 1.6 million tweets and 140 datasets. We employed different unsupervised and supervised algorithms, and we proposed a new clustering algorithm that we designed exclusively for this project. All the algorithms we use in the study can classify each tweet in the dataset into a cluster based on its characteristics and properties. The unsupervised clustering algorithm that we implemented includes K-means, mean shift, agglomerative hierarchical clustering, density-based spatial clustering of application with noise (DBSCAN), expectation-maximization clustering using gaussian mixture models (EMGMM), and latent dirichlet allocation unsupervised (LDA). Additionally, we proposed a custom supervised k-means algorithm that we evaluated for the first time using the same dataset to analyze its performance compared with standard unsupervised algorithms and techniques. For each clustering algorithm, we applied three different assessment metrics for clustering algorithms. This Included the Silhouette coefficient, Davies Bouldin index, and Dunn index to determine the cluster performance and accuracy of each clustering algorithm objectively.

1. INTRODUCTION

Twitter is one of the fastest-growing microblogging and online social networking sites in the world that enables users to send and receive messages in the form of tweets. Twitter is trending today in news and discussions. As of May 2020, about 500 million tweets are being generated every day, 350 tweets are sent per minute, and 200 billion tweets are created per year. Although Twitter

provides a list of the most popular topics people tweet about, it is often hard to understand what these trending topics are about and where they come from. Therefore, it is important to classify these topics into categories with high accuracy for a safer and more secure information retrieval. Automatic tools using machine learning techniques and algorithms allow the classification of tweets and datasets based on the content and keywords contained within. These tweets and datasets are organized into clusters. This type of data classification is very useful inside a social media platform or for research projects. This usefulness is true primarily if implemented along with other machine learning techniques like sentiment analysis [11]. For example, one of the possible applications of clustering techniques is that it allows the evaluation and classification of different data points into clusters in real time to determine if a tweet has violated the terms and conditions of a platform. Today, these kinds of cluster techniques are so advanced that they can train models to recognize and classify tweets or datasets into different clusters in a matter of milliseconds with high precision and accuracy.

Clustering algorithms, just like any other type of technology in society, have positive and negative effects. In addition to the adverse effects, machine learning techniques like clustering have raised many ethical concerns and questions among researchers and scholars around the world. These ethical concerns are based on past events where federal agencies and the government misuse this type of technology for personal benefits or for profiling [22]. In the worst-case scenario, it has even been used to draw assumptions and inferences against individuals based on what they post or write online. However, clustering also raises questions about privacy and the unauthorized collection and use of private data. It isn't difficult to find information that supports these ethics concerns. For example, the dataset used for this study was obtained from a very popular site that contains millions of datasets; this includes datasets that contain personal user information like email addresses and deleted tweets or texts. In most cases, this type of online platform collects user information such as tweets through API requests without informing or getting the user's permission. Once the data is collected in those platforms, anyone can access or download that dataset without providing any personal information or explanation about why they need that dataset in the first place, which can prove frightening for many people [4].

The intention of this study is not to try to discount the benefits of these types of technologies and techniques with the arguments presented, but rather to make the case that the development and use of these types of algorithms and datasets require regulations and oversight from developers, private agencies, and the government. The argument that datasets like the one used in this project need to be publicly available to develop and test new techniques and models that can improve current technologies making them better, faster, and easier to use does not justify the possible implications that this kind of technology can have on someone's privacy and life. This Includes the ethical concerns surrounding the misuse of this kind of technique and datasets by bad actors [5] [8]. For that reason, we decided to explore and investigate the current ethical challenges of clustering algorithms. At the same time, we measure the clustering performance of different supervised and unsupervised clustering algorithms using different assessment metrics. This type of analysis is relevant because it provides an accurate idea of how machine learning clustering algorithms and techniques could perform in real-world situations. Also, it can provide us insights into how accurately and effectively these kinds of techniques and algorithms are classifying different data points without any information about the dataset in different scenarios and applications. The results of this study

can be helpful for future implementations and developments of new algorithms and techniques that will allow for grouping different data points into clusters more quickly and with precision. More importantly, we want to use this project to illustrate and bring awareness about the ethical concerns around this kind of technology. Like most of us reading in this field, we want to see changes regarding the use, collection, and sharing of user data by private companies and government agencies. We understand that it's impossible to address all the ethical concerns and questions in this kind of technology. Still, we cannot ignore it and pretend that those ethical concerns and adverse effects do not exist. Hopefully the government, private companies, and developers can work together to tackle these issues. With some regulations, we believe that in near future, we can assure all citizens and users that their online information is not being used without their permission for machine learning projects [7]. Most importantly, we can guarantee that the development of future machine learning techniques is not being used for the wrong purpose of targeting, discriminating, or profiling individuals [10] [12].

2. CLUSTERING ALGORITHMS

Clustering algorithms have been around for many decades, since the 19th century, when the idea of k-means clustering was proposed for the first time by someone named Steinhaus. As society and technology evolved, clustering algorithms improved, incorporating principles and techniques from machine learning and artificial intelligence. Clustering is an unsupervised machine learning technique that involves the grouping of data points. It is a common technique for statistical data analysis used in many fields. Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group. In this project, we proposed a new supervised clustering technique and seven well-known supervised and unsupervised clustering techniques. We did this to study the performance and accuracy of this type of algorithm under the same set of data points. For simplicity of the project, all the algorithms except for the K-means supervised learning algorithm were implemented using pre-existing libraries and packages that can be found on the latest python version. Those libraries and packages were used to pre-process the data and then apply the different clustering techniques. These clustering algorithms include custom k-means supervised learning, k-means unsupervised learning, mean shift, agglomerative hierarchical clustering, density-based spatial clustering of application with noise (DBSCAN), expectation-maximization clustering using gaussian mixture models (EMGMM), and latent dirichlet allocation unsupervised (LDA).

2.1 K-MEANS CLUSTERING

The first technique that we explore in this project is K-means. K-means is probably the most well-known unsupervised clustering algorithm used by developers and researchers. Like any other unsupervised machine learning algorithms, k-means make inferences from datasets using only input vectors without referring to known, or labeled, outcomes. To process the learning data, the K-means algorithm starts with the first group of randomly selected centroids, which are used as the beginning points for every cluster. It then performs iterative calculations to optimize the positions of the centroids. The algorithm halts creating and optimizing clusters when either the centroids have stabilized and shown that there is no change in their values because the clustering has been successful or when the defined number of iterations has been achieved.

To implement the K-means unsupervised algorithm, we imported the sklearn library for K-Means and called the K-Means function with the number of clusters that we wanted as the argument. In this experiment, the number of the cluster that we used was three. K-Means has an advantage to other programs. It Is fairly quick, as all that is done is the computation of the distances between points and group centers, with few steps. However, the developer needs to manually define the number of classes and groups for the experiment. Thus K-means isn't always Ideal. With a clustering algorithm, it's important that the algorithm does the work to figure out the number of groups for us because the point is to gain some insight from the data. K-means also starts with a random choice of cluster centers, and therefore it may yield different clustering results on different runs of the algorithm. Thus, the results may not be repeatable and may lack consistency.

2.2 MEAN SHIFT

Another well-known unsupervised clustering algorithm is the mean shift. Mean shift clustering is an unsupervised sliding-window-based algorithm that attempts to find dense areas of data points. Like the k-means technique, mean shift is a centroid-based algorithm meaning that the goal is to locate the center points of each group, which works by updating candidates for center points to be the mean of the points within the sliding window. These candidate windows are then filtered in a post-processing stage to eliminate near-duplicates, forming the center points' final set and corresponding groups. Implementing mean shift clustering algorithm for this project was a very straightforward task, all that was required was to import the sklearn library for mean shift. In contrast to K-means clustering, there is no need to select the number of clusters as mean-shift automatically discovers this. That's a massive advantage. The fact that the cluster centers converge towards the points of maximum density is also quite desirable as it is quite intuitive to understand and fits in well in a naturally data-driven sense.

2.3 AGGLOMERATIVE HIRERCHICAL CLUSTERING

All the clustering algorithms that we study in this project use some type of attribute or property to classify elements into different clusters. Agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on similarity. It's also known as AGNES (Agglomerative Nesting). The algorithm starts by treating each object as a singleton cluster. Each pair of clusters are successively merged until all clusters have been incorporated into one big cluster containing all objects. The result is a tree-based representation of the objects, named dendrogram. Once this process is complete, the algorithm begins calculating the dissimilarity between the N objects using the agglomeration criterion. Two objects are clustered together, creating a new class comprised of these two objects. This is done only when it minimizes the agglomeration criterion. This process continues until all the objects have clustered. Using this type of iterative classification method has its advantages. For example, it works for the dissimilarities between the objects to be grouped together. A type of dissimilarity can be suited to the subject studied and the nature of the data. One of the results is the dendrogram which shows the progressive grouping of the data. It is then possible to gain an idea of a suitable number of classes into which the data can be grouped.

To implement the Agglomerative clustering, we imported the sklearn library for agglomerative clustering. We created an instance of TfidVectorizer, and then we transformed the data to tf-idf. Once the process was completed, we reduced the cluster centers to 2D using the reduced_feautures function from the sklearn package. Hierarchical clustering algorithms fall into two categories: top-down or bottom-up. For this study, we use the bottom-up algorithm in which we treat each data point as a single cluster at the outset and then successively merge pairs of clusters until all clusters have been merged into a single cluster that contains all data points. Additionally, the algorithm is not sensitive to the choice of distance metric; all of them tend to work equally well. However, with the other clustering algorithms, the choice of distance metric is critical. A particularly good case of using hierarchical clustering methods is when the underlying data has a hierarchical structure, and we want to recover the hierarchy

2.4 DENSITY BASED SPATIAL CLUSTERING WITH NOISE (DBSCAN)

From the clustering techniques that we covered so far in this paper, we can observe that despite the controversies and ethical concerns about the use and development of clustering algorithms, there is no denying that this type of technique has instrumental implementation. Clustering has been a critical tool for fields such as data analytics, data mining, and machine learning. For that reason, developers and researchers all around the world are working non-stop to adopt these machine learning clustering techniques to different areas of study in our society. For this reason, engineers have developed a clustering technique called density-based spatial clustering of applications with noise (DBSCAN). This clustering technique is relatively new but has significant properties and ethical challenges that we will discuss and study further in this paper. DBSCAN stands for density-based spatial clustering of applications with noise. To implement the DBSCAN algorithm, first, we defined the epsilon and min_sample values. For this experiment, our epsilon value is 3, and the min_sample value is 1, which represents the minimum number of points required to constitute a cluster. We imported three different packages from the sklearn library. The packages that we imported were metrics, make_circles, and DBSCAN. Once we had the packages inside the project, we called the DBSCAN function on the dataset.

Density based spatial clustering with noise technique is a well-known data clustering algorithm that is commonly used in data mining and machine learning. The algorithm can find arbitrary-shaped clusters and clusters with noise (i.e., outliers). The main idea behind DBSCAN is that a point belongs to a cluster if it is close to many points from that cluster. This clustery technique has a high-performance rate for a dataset where clusters have the constant density of data points. The cluster of random shape and size in large datasets with noise can be identified with significant accuracy. This is possible because one of the main attributes or advantages of this type of clustering algorithm is its noise cancellation. However, DBSCAN demonstrates reduced performances for clusters with different densities. Unlike the previous clustering algorithms, DBSCAN groups together data points that are close to each other based on the distance measurement, usually using the Euclidean distance formula as shown in figure 1.

Euclidean Distance (vectors) : $d(a, b) = \sqrt{\sum_i (a_i - b_i)^2}$

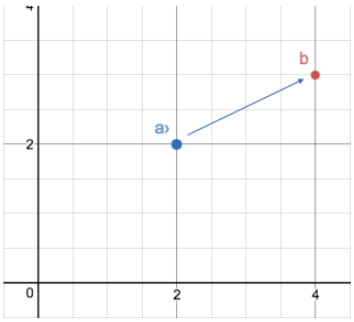

FIGURE 1: EUCLIDEAN DISTANCE

2.5 EXPECTATION MAXIMIZATION CLUSTERING USING GAUSSIAN MIXTURE MODELS (EMGMM)

Ethical questions surrounding new techniques and algorithms are very common, especially in machine learning and artificial intelligence. However, engineers and scientists alike cannot just stop developing new techniques and algorithms such as clustering because of fear or concern about its use and possible consequences on society. It is similar to the way that humans cannot control evolution. We cannot control the surge or rise of new technologies and algorithms. As a society powered by technology, we need to understand that there is no issue in the development of new technologies, because in most cases new scientific and technological developments are made with good intentions to improve society regarding a specific issue. The real problem with the ethics of algorithms is the lack of oversight and laws to regulate and prevent the misuse of these scientific and technological advances. More importantly, the problem is with the absence of legislation to protect the unauthorized collection, use, and share of users' private data by private companies, government, and individuals. It's mind-blowing how easy it is to find and download datasets that could potentially contain confidential and sensitive information about a person. If we lived in a perfect society where there were no wrongdoers, and everyone was honest and unbiased, this would not be an issue.

The last two algorithms that we explored were included in this study to illustrate and showcase some of these ethical concerns regarding data collection and what an individual with the right tools and knowledge can do with publicly available sensitive datasets like the one we used in this study. Gaussian mixture model (GMM) is a probabilistic model that assumes that all data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. To implement the EM Gaussian mixture model, we imported the GaussianMixture package from the sklearn library. Then we used the GaussianMixture function from the library. The first parameter of this function is the number of components that we wish to create. For this experiment, the number of components that we want is three. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the

centers of the latent Gaussians. EM-GMM algorithm implements the expectation-maximization (EM) algorithm for fitting mixture-of-Gaussian models. This proposed method is more suitable for finding arbitrary ellipsoidal clusters with a random number of data points. The GMM assigns query data points to the multivariate normal components that maximize the posterior component probability with the data given. That is, given a fitted GMM, the cluster assigns query data to the component yielding the highest posterior probability. This technique of clustering assigns a data point to exactly one cluster. GMM clustering can accommodate clusters that have different sizes and correlation structures within them. Therefore, in specific applications, GMM clustering can be more appropriate than methods such as k-means clustering. Like many clustering methods, GMM clustering requires the user to specify the number of clusters before fitting the model.

2.6 LATENT DIRICHLET ALLOCATION UNSUPERVISED (LDA)

Finally, the last clustering machine learning technique implemented is one of the most common models used in practice. Latent Dirichlet Allocation (LDA) is an unsupervised clustering algorithm where documents are assumed to be randomly generated by sampling topics from their topic mixture, and sampling words from those topics, and repeating this process to generate all words in the document. The topics are viewed as latent variables, and LDA executes by inferring the topics from the documents via a Dirichlet process. Topics are randomly seeded, and then iteration proceeds using Bayesian inference. To implement the latent dirichlet allocation unsupervised algorithm, first we imported the LatentDirich package from the sklearn library. Then we used the build in function LatentDirichLetAllocation from the library with two arguments. The first argument is three, which represent the number of components. The second argument is the number of jobs, which is just one. During each iteration, LDA compares each document with the topic and updates the topics for the next iteration, which continues until a stopping criterion is met. Convergence can be measured either by a change in the inferred parameters or by another objective metric of the model, such as the likelihood of producing the input set. The algorithm repeatedly samples the documents and modifies the topics to better fit them until reaching a specified convergence. LDA has several assumptions, including that both words and documents are unordered and that all documents are generated in the same time frame.

2.7 CUSTOM IMPLEMENATTION SUPERVISED K-MEANS CLUSTERING

Unlike an unsupervised learning algorithm, a supervised learning algorithm makes inferences from a dataset using only input vectors, without referring to known or labeled outcomes. Instead, this algorithm uses training data to analyze the dataset and train the model. Supervised clustering automatically adapts to the clustering algorithm with the aid of a training set consisting of item sets and complete partitioning of these item sets. This type of learning requires two sets of data. The first dataset is the training dataset. This dataset is used during the learning process to fit the parameters to train the model. The second dataset is the testing dataset. The testing dataset is used to provide an unbiased evaluation to the final model fit on the training data. For this paper, we proposed a new variant of k- means algorithm using the supervised learning technique. This k-means algorithm is an iterative clustering algorithm that aims to find local maxima in each iteration. After specifying the desired number of k clusters, the algorithm uses the training data to assign each data point to a cluster randomly. It computes cluster centroids and re-assigns each point to the closest cluster

centroid over and over again until no improvement is possible. Additionally, we provide a structural support vector machine (SSVM) algorithm for this supervised k-means learning algorithm. This structural support machine is capable of directly optimizing a parameterized similarity measure to maximize cluster accuracy.

3. ASSESSSMENT METRICS

All the algorithms that we assess in this paper, with the exception of the algorithm we proposed, do not require training data. This means that they are unsupervised machine learning techniques. We don't have any labels in clustering, just a set of features for observation. The goal is that clusters that have similar observations should be clubbed together and dissimilar observations between clusters should be kept as far as possible. Evaluating the performance of a clustering algorithm is not as trivial as counting the number of errors in an algorithm or the number of times the algorithm misclassifies an element. Here, clusters are evaluated based on some similarity or dissimilarity measure, such as the distance between cluster points. If the clustering algorithm separates dissimilar observations and puts similar observations together, then it has performed well. To accomplish that task and to be as unbiased and accurate as possible for the study. We decided to use three of the most popular metrics evaluations for clustering algorithms: Silhouette coefficient, Dunn's Index, and Davies Bouldin Index.

3.1 SILHOUETTE COEFFICIENT

Silhouette Coefficient or silhouette score is a metric used to calculate the quality of a clustering technique. The coefficient for a set of samples is given as the mean of the Silhouette coefficient for each sample as shown in figure 2. The first variable in the function *a* represents the mean distance between a sample and all other points in the same class. The second variable *b* represents the distance a sample and all other points in the nearest cluster. Its value ranges from -1 to 1. The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering. Scores around zero indicate overlapping clusters. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster. The Silhouette analysis is suitable for this study because it gives us an idea of how accurately the different clustering algorithms assign each tweet in the dataset into the different clusters. However, it can also be used to study and understand the separation distance between the resulting clusters.

$$s = \frac{b - a}{\max(a, b)}$$

FIGURE 2: SILHOUETTE COEFFICIENT FORMULA

3.2 DUNN INDEX

The second metric we used for the study is Dunn's index. Dunn's Index (DI) is another metric for evaluating a clustering algorithm. Dunn's Index is equal to the minimum inter-cluster distance

divided by the maximum cluster size. The Dunn index is defined as the quotient of d_{min} and d_{max} as shown in figure 3. Note that large inter-cluster distances (large separation) and smaller cluster sizes (more compact clusters) lead to a higher DI value. A higher DI implies better clustering. It assumes that better clustering means that clusters are compact and well-separated from other clusters. The Dunn index metric can help us determine the quality of the clustering structure. It can also be used to estimate the number of clusters on each algorithm.

$$d_{min} = \min_{k \neq k'} d_{kk'}$$

$$d_{max} = \max_{1 \leq k \leq K} D_k$$

FIGURE 3: DUNN INDEX

3.3 DAVIES BOULDIN INDEX

Finally, the last metric implemented in this study is the Davies Bouldin Index. The Davies Bouldin index is a validation metric that is often used to evaluate the optimal number of clusters to use. It is defined as a ratio between the cluster scatter, and the cluster's separation as shown in figure 4. A lower value will mean that the clustering is better. The Davies index can be used in the project to determine the optimal number of clusters that we need in each algorithm to produce the most accurate clustering classification for the dataset.

$$\mathcal{C} = \frac{1}{K} \sum_{k=1}^{K} M_k = \frac{1}{K} \sum_{k=1}^{K} \max_{k' \neq k} \left(\frac{\delta_k + \delta_{k'}}{\Delta_{kk'}} \right)$$

FIGURE 4: DAVIES BOULDIN INDEX

4. DATASETS

The dataset we used for this study is public and can be easily found and download online without permission. The dataset contains six fields for the sentiment of 1.6 million tweets extracted using the Twitter API without the consent or knowledge of the user. For this study, we prepared an unlabeled dataset of only tweets from the original dataset. Datasets like these raise some privacy and ethics concerns because the set contains sensitive and private information about an individual. In this case, some of the tweets found in the dataset are from users that had chosen to delete tweets from their account or from users who had closed their account many years ago. Those users are living their lives thinking that their data was removed already and that there are no traces or records of their tweets on the internet. Still, as we can observe in the data set, we can find the individual's

username, their ID number, and the exact tweet that was posted. Scenarios like this illustrate the ethical concern that many people have regarding social media platforms and how their data is being managed, used, or shared across different platforms even when they are not using the application anymore. Not only datasets like these raise ethical issues, but machine learning techniques like the clustering algorithms we described and utilized in this project are also a concern.

All clustering algorithms have the same goal to classify data into different categories. The duty of these algorithms seems inoffensive and harmless, but everything can and does have a dark side. As human beings, it's normal to be worried about our safety and the safety of the people around us. What many of us ignore the fact that not all the threats to our lives and security are visible. There are invisible threats around us without us even noticing them, and those are the ones we need to worry about and pay more attention to [7]. The invisible threads we are referring to are machine learning algorithms. We live in a complex world where every system is connected and is constantly interacting with everything around it—sharing and collecting data from all sources and platforms. Nothing we do is private anymore; for that reason, we need to know who has access to our data and how the entities who have access to our data are using it. Clustering algorithms can look harmless from the surface, but they can affect and possibly change many lives in a matter of seconds. For example, the government or social media platforms can use clustering algorithms to make inferences about an individual [3] [6]. Those inferences can be true or false, but the algorithm can make it appear true because all the data that was collected about the individual suggests that and is grouped together. A situation like this raises many ethical questions and concerns about the use of this type of machine learning algorithm to profile and make assumptions about an individual without his/her knowledge.

Companies like Google or Facebook can use the result of these algorithms to target specific ads to individuals, similar to what we saw in the 2016 presidential elections with Cambridge Analytica, where millions of pieces of Facebook user data was obtained and used without permission for political advertising. A situation like this can occur using clustering algorithms to infer someone's political preference and then show specific political ads, propaganda, or information to influence the person's ideals and beliefs [5]. To understand how powerful and dangerous these clustering algorithms are, in the next section, we are going to apply those algorithms with the dataset. It is Important to find out how accurate the algorithms can be in classifying data that could potentially be used to make inferences about a person.

5. EXPERIMENT

For this study, to make the process simple and easy to follow, we have divided the experiment into three steps. The first step was to pre-process the dataset. This pre-process included cleaning and filtering the data to the correct format for the clustering algorithms. The second step of the experiment was to run the two datasets individually on each clustering algorithm and then measure the results using three different assessment metrics. The last step was to record the result obtained on the clustering algorithm and the assessment metric.

We downloaded the recent dataset of sentiment tweets. This contained 1.6 million tweets and 140 datasets during the first step of the experiment. We then prepared two unlabeled datasets—

the first data set with 10,000 tweets and the second dataset with 50,000 tweets. The tweets on the new unlabeled datasets that we created contain unnecessary objects such as hashtags, mentions, links, and punctuation that can affect the performance of the clustering algorithms. We used different libraries like TfidVectorizer and custom functions to sanitize and remove the unnecessary characters on each tweet in the datasets. Once that process was complete, we converted all the text in the tweets to lower case to avoid causing the different algorithms to interpret words with different cases as different. In the last step, to clean the tweets before feeding them into the various clustering algorithms for testing, we decided to split each tweet into a list of tokens or words. After this, we reduced each word to its root form using NLP and the lemmatizer library. Finally, we removed all stop words that have less weight compared to other words in the tweet. This includes words like 'and', 'or', 'has', etc.

After preprocessing the dataset and cleaning the tweets, we ran the new dataset of tweets on each of the different clustering algorithms discussed above in this paper, including the new algorithm that we proposed. To start, the first clustering algorithm that we tested was k-means. In the k-means experiment, we chose three clusters. We chose to use three clusters because we did not want to over fit the model. However, at the same time it was important to make sure that we had enough clusters possible so that the algorithm could still differentiate and group similar clusters together. Once we defined the number of clusters inside the main function of the code, we executed the k-means algorithm on the two data sets, recorded the results, and graphed the clusters using the math plot library for easy visualization. Once the results were recorded and the clusters were defined, we took the results and passed them to the different assessment metrics functions to measure the cluster performance and accuracy. We considered repeating the experiment multiple times and getting the average of each test as the final result. However, we decided not to do this because k-means is a non-deterministic technique, meaning that each point in the data is randomly assigned to a cluster at the beginning. For that reason, each program's execution will give us a different result. We wanted to get the most authentic and accurate results of the clustering algorithm performance without adding any external factors or variables that could alter the results.

The next clustering algorithm that we tested was the mean split. Unlike k-means, in this experiment we did not have to specify in the code the number of clusters that we wanted to receive. The algorithm automatically assigned and found the number of clusters based on the datasets. We executed the mean-split algorithm on the two datasets. After using the results of the clustering, we measured the cluster performance by running the three different assessment metrics programs on the clustering result. In the end, we recorded and graphed the results for the clustering algorithm and the assessment metric. At this point of the study, we wanted to speed up the process and evaluate multiple algorithms simultaneously. We decided to put the remaining five unsupervised clustering algorithms (Agglomerative Hierarchical Clustering, Density-Based Spatial Clustering with Noise, Maximization Clustering Using Gaussian Mixture Models, and Latent Dirichlet Allocation Unsupervised) into a single program that will execute each cluster process individually with an exact copy of each dataset. The results of each clustering technique were recorded in separate files and then analyzed by all three-assessment metrics for clustering performance and accuracy. The result of each assessment metric was graphed individually using math plot python libraries and packages.

Finally, for the last experiment, we tested our proposed clustering algorithm. Unlike the previous algorithms that we tested in this paper, this new algorithm required supervised learning, meaning it required training and testing data to train and test the model. For this experiment, we divided both data sets into two equal batches as shown in figure 19 and 20. To achieve this, we randomly split each data set into two different JSON files. In the end on this step, we had four data sets in total—two data sets of 5,000 tweets and two data sets of 25,000 tweets. The first pair of each data set is for training, and the second pair for testing. During the training phase of the experiment, we fed the algorithm each training dataset to train the model of the algorithm. Once that process was finalized, we provided the algorithm with each testing data set. The testing phase results were recorded and graphed using math plot to generate a 3d graph to illustrate the results. Finally, the result of the algorithm was used to calculate the accuracy of the model using the three different assessment metric programs. Once all the above steps were completed successfully, we created a bar graph using python and math plot library to generate a diagram to illustrate the results of the different assessment metrics on each of the seven different algorithms that we tested.

6. EXPERIMENT RESULTS

In this section, the overall effectiveness of the clustering algorithm is calculated using three different metric assessment metrics: Davies Bouldin index, Dunn index, and silhouette coefficient. The overall accuracy measurement determines how well the clustering algorithm is able to create clusters that contain different tweets.

In the following subsections, the effectiveness of the K-means, mean shift, agglomerative hierarchical clustering, density-based spatial clustering of application with noise (DBSCAN), expectation-maximization clustering using gaussian mixture models (EMGMM), latent dirichlet allocation unsupervised (LDA), and supervised K-means algorithms are presented.

6.1 K-MEANS CLUSTERING

The K-means algorithm has an input parameter of K. This input parameter, as mentioned in section 2.1, is the number of clusters used by K-means. K-means algorithm was evaluated with K equal 3 in each of data set that we have. The result of the clustering algorithm on 10000 dataset is shown in figure 21. The overall accuracy of K-means using the metric assessments is approximately 2.2 for Davies Bouldin index, 0.2 for Dunn index, and 0.25 for silhouette coefficient. However, when we examined the results on the 50000 dataset, we observed a slight improvement on the Davies Bouldin index as shown in figure 22.

As we stated in section 3.2 and 3.1, a higher Dunn index and silhouette coefficient value indicates a good clustering. This means that based on the result of these two indexes in both data set, our implementation of K-means did not classify the tweets correctly into the three different clusters that we defined. However, when we examined the accuracy of the same algorithm using the Davies Bouldin index on both datasets, it shows that the implementation of k-means on the data set was able to classify the different tweets into their corresponding cluster accurately.

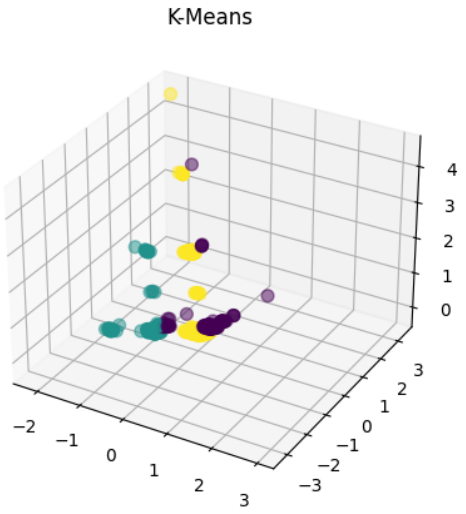

FIGURE 5: K-MEANS UNSUPERVISED 10000

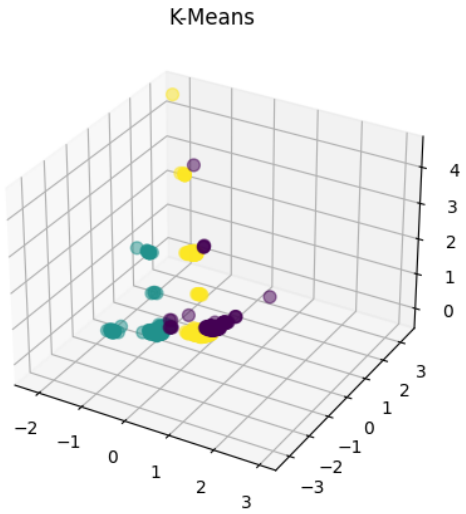

FIGURE 6: K-MEANS UNSUPERVISED 50000

6.1.2 MEAN SHIFT

Unlike k-means, there is no need to select the number of clusters in mean shift, since it automatically discovers this value for this parameter. The results for the agglomerative hierarchical clustering for each data set are shown in figure 21 and 22. Overall, the mean shift algorithm has the lowest accuracy for the Silhouette coefficient. It does however have the highest accuracy for Davies Bouldin index, which is a good indication of good clustering and performance because as we stated in section 3.3, a lower value for Davies Bouldin index means that the clustering is better.

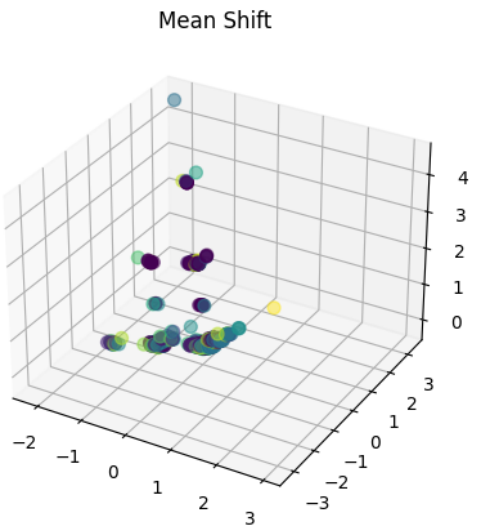

FIGURE 7: MEAN SHIFT 10000

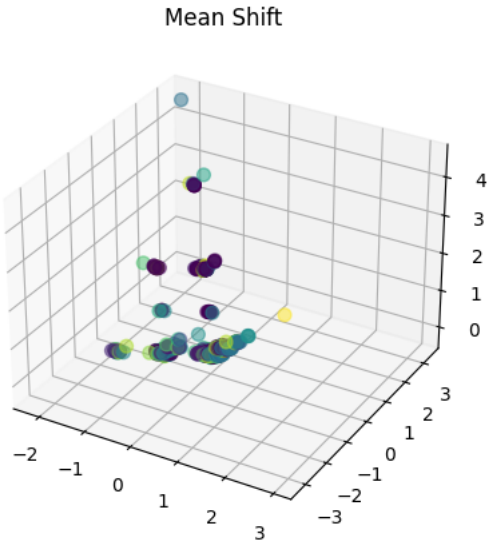

FIGURE 8: MEAN SHIFT 50000

6.1.3 AGGLOMERATIVE HIRARCHICAL CLUSTERING

The result for the agglomerative hierarchical clustering for both experiment is shown in figure 21 and 22. For this algorithm, the number of clusters and the cluster parameters are automatically determined. The agglomerative hierarchical algorithm has a value of 0.2 for the Dunn index and 0.32 for the silhouette coefficient in the first experiment, which indicates a low accuracy for clustering. With the Davies Bouldin index, the score indicates a good clustering performance when the index score was compared with the first two algorithms (K-mean and mean shift). In the second experiment we can observed a slight improvement in Dunn index and Silhouette coefficient.

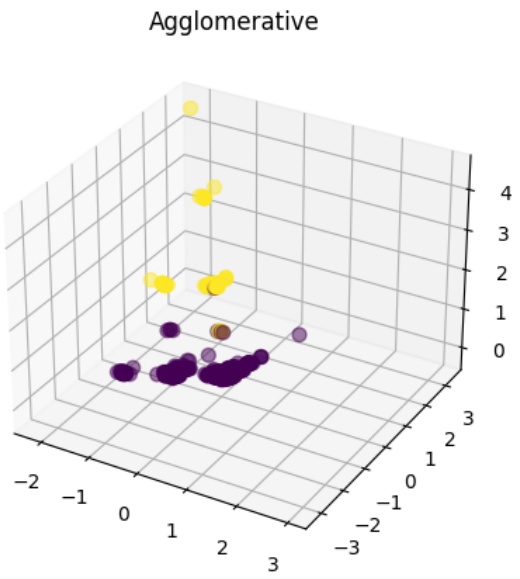

FIGURE 9: AGGLOMERATIVE 10000

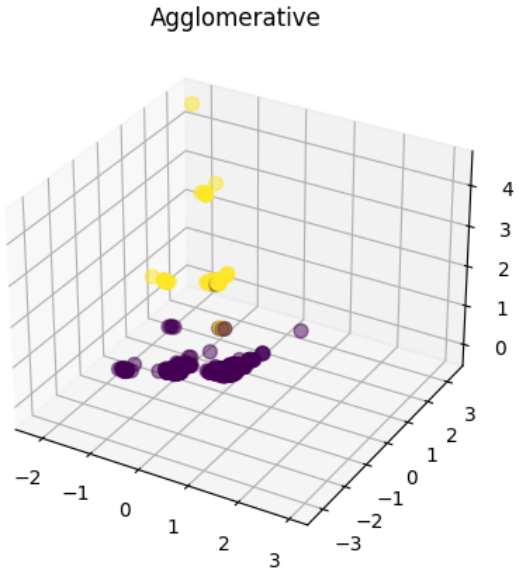

FIGURE 10: AGGLOMERATIVE 50000

6.1.4 DENSITY-BASED SPATIAL CLUSTERING OF APPLICATION WITH NOISE (DBSCAN)

The accuracy results for the DBSCAN algorithm are presented in Figure 21 and 22. Recall that DBSCAN has two input parameters. We did not vary these parameters. Instead, we defined those values as fixed variables in the algorithm. The values used for minPts was 40, and the eps distance was 0.003. The accuracy of the algorithm on both datasets were very similar to the results we obtained on the k-means experiment. Dunn index and silhouette coefficient index indicated a low accuracy for the algorithm meaning a bad clustering performance. However, Davies Bouldin index suggested that the algorithm performed exceptionally well in classifying the tweets into the clusters, but the performance and accuracy were not as high when compared with the K-means algorithm.

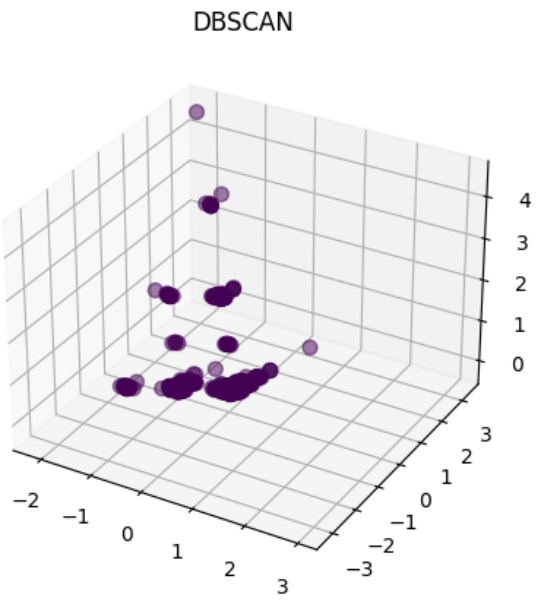

FIGURE 11: DBSCAN 10000

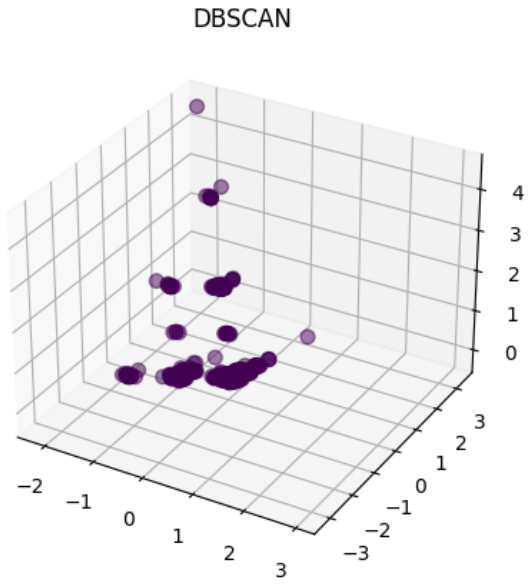

FIGURE 12: DBSCAN 50000

6.1.5 EXPECTATION-CLUSTERING USING GAUSSIAN MISTURE MODELS (EMGMM)

The results for the expectation maximization clustering using gaussian mixture models (EMGMM) are shown in figure 21 and 22. For this algorithm, the model assumes that all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Overall, in both experiments the expectation-maximization algorithm has the lowest accuracy for all three metric assessments for clustering algorithms. The results indicate a low clustering performance. It has a value of 8.2 for the Davies Bouldin index, 0.2 for the silhouette coefficient, and 0.25 for the Dunn index.

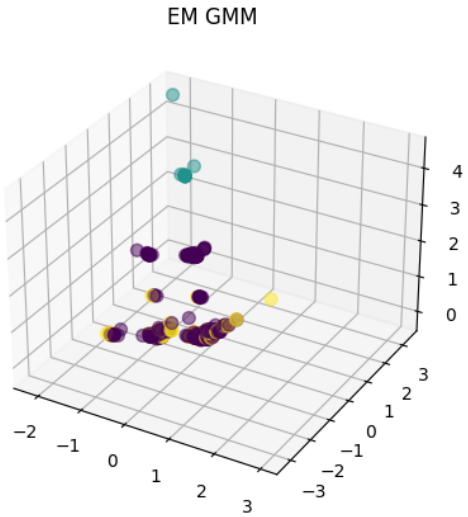

FIGURE 13: EM GMM 10000

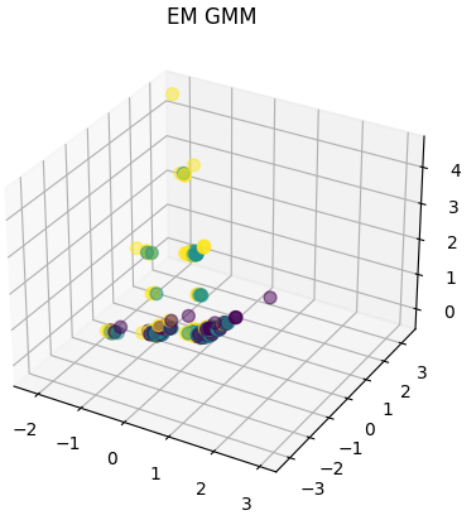

FIGURE 14: EM GMM 50000

6.1.6 LATENT DIRICHLET ALLOCATION UNSUPERVISED (LDA)

LDA clustering algorithm makes several assumptions. These assumptions Infer that both words and documents are unordered and that all documents are generated in the same time frame. The score for the Dunn index was close to 0 in both experiment (10000 and 50000 dataset) as shown in figure 21 and 22, which indicates that it completely fails in clustering the tweets in the correct clusters. However, at the same time it showed the best performance and accuracy among all the algorithms using the Davies Bouldin Index and silhouette coefficient.

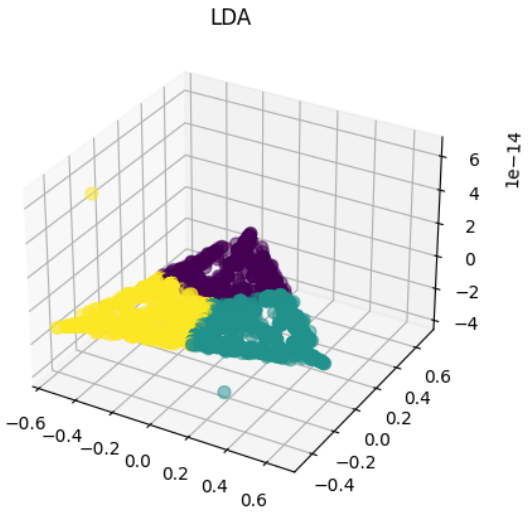

FIGURE 15: LDA 10000

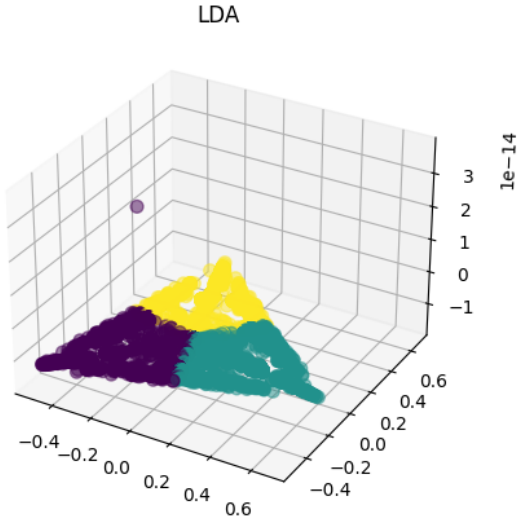

FIGURE 16: LDA 50000

6.1.7 CUSTOM K-MEANS SUPERVISED CLUSTERING

The overall accuracy of this supervised learning k-means algorithm steadily improves when compared with the accuracy of the unsupervised version of the same algorithm. We found that the accuracy of the algorithm improves in all three metric assessments for clustering algorithms in both datasets as shown in figure 21 and 22. A score of 0.35 for the Dunn Index and 0.34 for the Silhouette coefficient indicates that the algorithm was average, and that it was able to cluster the tweet with some degree of accuracy. Upon examining the results, we can see a + 0.1 improvement in the Silhouette coefficient and Dunn Index from the unsupervised k-means algorithm and a -0.5 in the Davies building index in both datasets. This regression of the Davies Bouldin index indicates a better cluster accuracy and performance.

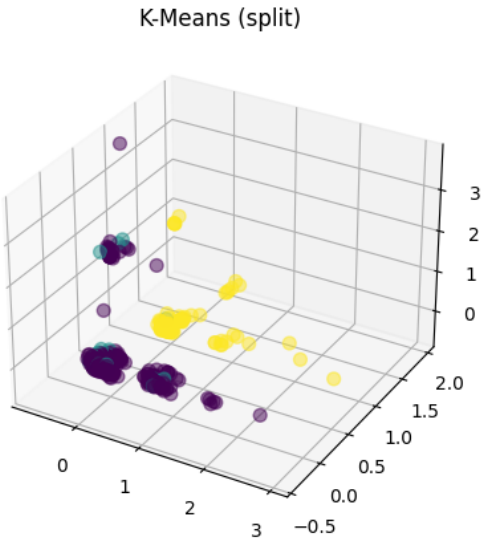

FIGURE 17: K-MEANS SUPERVISED 10000

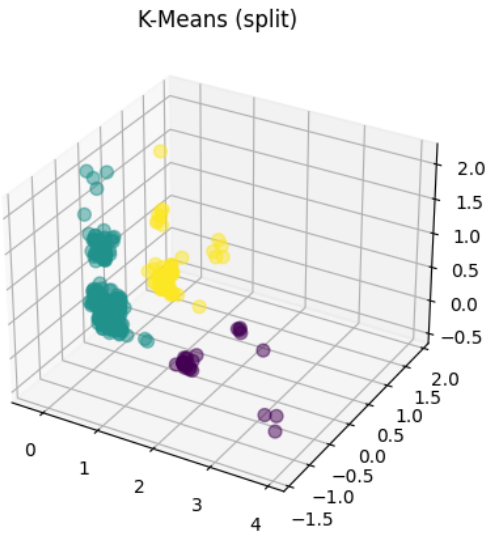

FIGURE 18: K-MEANS SUPERVISED 50000

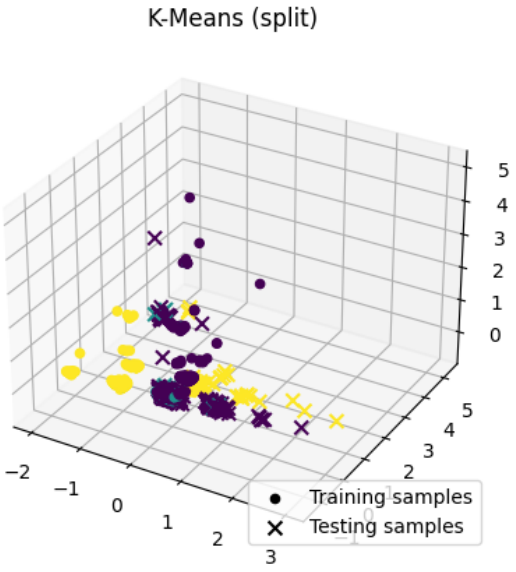

FIGURE 19: K-MEANS BATCH 10000

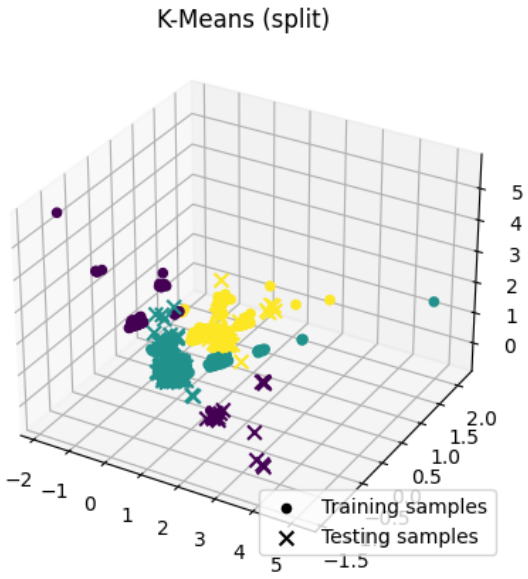

FIGURE 20: K-MEANS BATCH 50000

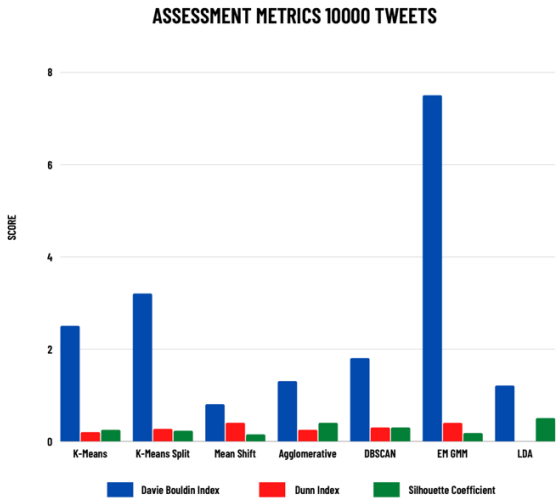

FIGURE 21: ASSESSMENT METRIC 10000

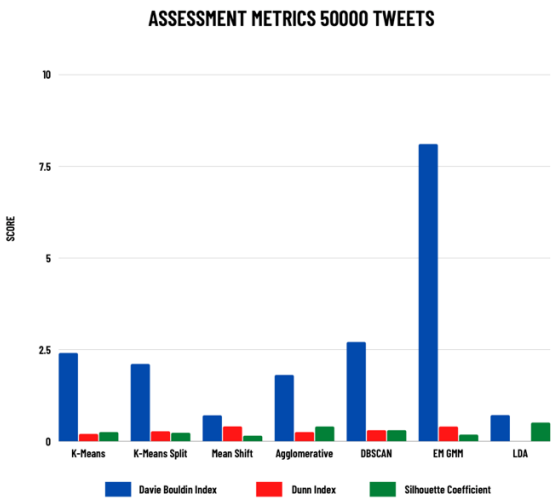

FIGURE 22: ASSESSMENT METRIC 50000

7. CONCLUSION

In this paper, we evaluated eight different clustering algorithms. These were k-means, mean shift, agglomerative hierarchical clustering, density-based spatial clustering of application with noise (DBSCAN), expectation-maximization clustering using gaussian mixture models (EMGMM), latent dirichlet allocation unsupervised (LDA), and custom supervised K-means, for the clustering prediction based on clustering prediction tweet problem. Our main goal with the experiment was to illustrate the ethical concerns and questions about the use of this type of machine learning algorithm. Our analysis of the different clustering algorithms shows that this kind of algorithm provides great benefits to our society in areas like data analysis, data mining, and research projects in all fields. This would include, for example, classifying and analyzing DNA sequences and proteins into different groups to identify genes that cause specific diseases. It has many positive traits to bring to society, but at the same time raises a lot of ethical problems that are hard to ignore.

The first ethical issue that we can observe from this experience is the easy access to online datasets that contain personal information about a person. These types of data sets are available online for anyone to use or download without any regulation or oversight. This is a big deal because not only are companies and individuals collecting user data online without permission and knowledge of the party involved, but sites that provide this information are also allowing anyone to obtain and use data that sometimes includes sensitive information about individuals [2] [11]. In a world like the one we live in, it can be inferred that people with bad intentions will use these data sets to commit crimes.

The second ethical concern is that individuals or organizations misuse this type of machine learning algorithm to profile, discriminate, or make assumptions about an individual without verifying that those assumptions are accurate. This may seem harmless and inoffensive on the surface, but this kind of practice can be a constant threat to individuals. For example, companies like

Google or Facebook can use clustering algorithms to make inferences about someone's political party based on their online activity on social media. They can and already do use that information without knowing if it's accurate or not to target specific ads, posts, news, or videos to influence whoever is seeing it on their decisions and beliefs [5] [7]. This targeted advertising is similar to what we saw in the 2016 presidential elections with Cambridge Analytica, where millions of Facebook users' data was obtained and used without permission for political advertising.

Our intention is not to try to discount the benefits of these kinds of technologies and techniques with the arguments presented. Rather, we wish to make the case that the development and use of these kinds of algorithms and datasets require regulations and oversight from developers, private agencies, and the government. We wanted to illustrate and bring awareness about the ethical concerns around this kind of technology. Like most of us reading in this field, we want to see changes regarding the use, collection, and sharing of user data by private companies and government agencies. We understand that it's impossible to fix all the ethical concerns and questions in this kind of technology, but we cannot just ignore the problems and pretend those ethical concerns and adverse effects do not exist. Ours is a work in progress. We will continue to investigate these and other clustering algorithms for use as an efficient classification tool and to analyze the ethical implications of these types of technologies in our society.

8. REFERENCES

1. Antenucci, Dolan, et al. "Classification of tweets via clustering ofhashtags." EECS 545 (2011): 1-11.
2. The Ethical Challenges of Publishing Twitter Data for Research Dissemination. ResearchGate. (n.d.).
3. Webb, H., Jirotka, M., Stahl, B. C., Housley, W., Edwards, A., Williams, M., ... Burnap, P. (2017). "The Ethical Challenges of Publishing Twitter Data for Research Dissemination." *Proceedings of the 2017 ACM on Web Science Conference*.
4. Soman, Saini Jacob, and S. Murugappan. "Detecting malicious tweets in trending topics using clustering and classification." 2014 International Conference on Recent Trends in Information Technology. IEEE, 2014.
5. Martin, Kirsten E. "Designing ethical algorithms." MIS Quarterly Executive June (2019).
6. Emmons, Scott, et al. "Analysis of network clustering algorithms and clusterquality metrics at scale." PloS one 11.7 (2016): e0159161.
7. Mittelstadt, Brent Daniel, et al. “The Ethics of Algorithms: Mapping the Debate.” Big Data & Society, vol. 3, no. 2, 2016, p. 205395171667967.,doi:10.1177/2053951716679679.
8. Sisodia, Deepti, et al. "Clustering techniques: a brief survey of different clustering algorithms." International Journal of Latest Trends in Engineering and Technology (IJLTET) 1.3 (2012): 82-87.
9. Qian, Wei-ning, and A. Y. Zhou. "Analyzing popular clustering algorithms from different viewpoints." Journal of software 13.8 (2002): 1382-1394.
10. Awad, Edmond, et al. "The moral machine experiment." Nature 563.7729(2018): 59-64.
11. Clustering of Tweets: A Novel Approach to Label the Unlabelled Tweets
12. Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. 2019. Clustering without Over-Representation.
13. Aris Anagnostopoulos, Luca Becchetti, Matteo Bo ̈hm, Adriano Fazzone, Stefano Leonardi, Cristina Menghini, and Chris Schwiegelshohn. 2019. Principal
14. Fairness: Removing Bias via Projections.

15. Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. 2019. Scalable fair clustering
16. Suman K Bera, Deeparnab Chakrabarty, and Maryam Negahbani. 2019. Fair algorithms for clustering.
17. Xingyu Chen, Brandon Fain, Charles Lyu, and Kamesh Munagala. 2019. Proportionally Fair Clustering.
18. Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair clustering through fairlets.
19. Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017.
20. Fair clustering through fairlets.
21. Dheeru Dua and Casey Graff. 2017. UCI machine learning repository (2017).
22. Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In Proceedings of the 3rd innovations in theoretical computer science conference.
23. Anil K Jain. 2010. Data clustering: 50 years beyond K-means. Pattern recognition letters 31, 8 (2010), 651–666.
24. Anil K Jain, Richard C Dubes, et al. 1988. Algorithms for clustering data. Vol. 6. Prentice hall Englewood Cliffs.
25. Prateek Jain, Raghu Meka, and Inderjit S Dhillon. 2008. Simultaneous unsupervised learning of disparate clusterings. Statistical Analysis and Data Mining: The ASA Data Science Journal 1, 3 (2008), 195–210.
26. Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In International Conference on Computer, Control and Communication. 1–6.
27. Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 2019. Fair k-center clustering for data summarization.
28. Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. 2019. Guarantees for spectral clustering with fairness constraints.
29. Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In International conference on machine learning. 1188–1196.
30. James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1. Oakland, CA, USA, 281–297.
31. Matt Olfat and Anil Aswani. 2019. Convex formulations for fair principal component analysis. In AAAI, Vol. 33. 663–670.
32. Clemens Rösner and Melanie Schmidt. 2018. Privacy preserving clustering with constraints.
33. Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20 (1987), 53–65.
34. Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. 2018.Fair coresets and streaming algorithms for fair k-means clustering.
35. Bokun Wang and Ian Davidson. 2019. Towards Fair Deep Clustering With Multi-State Protected Variables.
36. Imtiaz Masud Ziko, Eric Granger, Jing Yuan, and Ismail Ben Ayed. 2019. Clustering with Fairness Constraints: A Flexible and Scalable Approach.