Comparison of Clustering Algorithms Under Different Conditions

Eren Özdemir[1], Muhammet Musa Çam[2], Onur Orkun Kader[3]

1)eren.ozdemir@st.bau.edu.tr
Department of Comp. Eng.

2)muhammetmusa.cam@st.bau.edu.tr
Department of Comp. Eng.

3)onurorkun.kader@st.bau.edu.tr
Department of Comp. Eng.

Abstract

Sometimes it is difficult to extract meaningful results from datasets. Whether they are too inconsistent or there are scarce. With the development of machine learning, there will be always new approachment to this problem. In this paper, we aimed that how can they differ one from another under different clustering algorithm mechanism. There are some popular clustering algorithms such as K-Means clustering algorithm and Agglomerative clustering algorithm. Our studies shows the difference between those clustering algorithms and the other ones which is not popular to use such as Birch clustering algorithms and OPTICS clustering algorithms. We implemented 9 different clustering algorithms to two different datasets in order to see difference between small data and large data.

Keywords: Cluster, algorithms, machine learning, k-means, Agglomerative, Birch, OPTICS

1. Introduction

Data clustering, also known as data segmentation, seeks to divide a set of data into a certain number of subsets (or clusters) that are optimal according to a set of criteria.

In order to discuss clustering, we have to mention one machine learning algorithm. Unsupervised learning. Unsupervised learning uses machine learning algorithms to examine and cluster unlabeled dataset. Unsupervised learning has different main tasks and clustering is one of them. In clustering algorithm, for example, K-Means clustering algorithm assign similar data points into one group where is K represented by the number of groups in the dataset chunks. As mentioned in [3] the clustering algorithms must have following properties:

a) Data objects within the cluster must be like or near to each other as much as possible.
b) Data objects belong to different clusters must be dissimilar or far off to each other as much as possible.
c) The distance / similarity measure must have some practical ability and be clear.

Thanks to these features, clustering is also used in fields such as image segmentation, image pattern recognition, object recognition, information retrieval, and bioinformatics. [4]

Finding and using the appropriate clustering algorithm has also been a matter of concern recently. As an example, K-Means is a popular algorithm to find clusters in different datasets but it is weak when working with non-spherical clusters data points. As another example, hierarchical clustering can produce more successful results and better visualization than other algorithms but it requires more computational power. In [5], [6], [7] related works for comparing different clustering algorithms made. 9 different clustering algorithms are fruitfully compared in this paper on 2 different datasets. Star-Classification dataset [1] is partially small

dataset which contains 240 row. E-Commerce Shipping dataset [2] is partially big dataset which contains over 10k row in itself. Our main goal here is to observe how clustering algorithms differ in small and large datasets and how much the performance of the algorithms will change depending on the spread of the data on the coordinate plane. Section 1 introduces need and explaining of comparison of clustering algorithms on different datasets. Section 2 explains the dataset we used during this paper. Section 3 describes the methods we used for this study. Section 4 shows the obtained results from the study. Section 5 summarizes the conclusions about clustering algorithms, their advantage and disadvantage over the small data and large data.

2. Dataset Description

1) Star-Type Classification Dataset

Context: Star Classification dataset and their related components whether they are belongs to dwarf group or giants group.

Content:

- **Temperature** - K
- **Relative Luminosity** - L/Lo
- **Relative Radius** - R/Ro
- **Absolute Magnitude** - Mv
- **Color** - General Color of Spectrum
- **Spectral_Class** - O,B,A,F,G,K,M / SMASS [8]
- **Type (0 to 5)** - Red Dwarf, Brown Dwarf, White Dwarf, Main Sequence, Super Giants, Hyper Giants

2) E-Commerce Shipping Dataset

Context: An international e-commerce company based wants to discover key insights from their customer database. They want to use some of the most advanced machine learning techniques to study their customers. The company sells electronic products.

Content: The dataset used for model building contained 10999 observations of 12 variables. The data contains the following information:

- **ID:** ID Number of Customers.
- **Warehouse block:** The Company have big Warehouse which is divided in to block such as A,B,C,D,E.
- **Mode of shipment:**The Company Ships the products in multiple way such as Ship, Flight and Road.
- **Customer care calls:** The number of calls made from enquiry for enquiry of the shipment.
- **Customer rating:** The company has rated from every customer. 1 is the lowest (Worst), 5 is the highest (Best).
- **Cost of the product:** Cost of the Product in US Dollars.
- **Prior purchases:** The Number of Prior Purchase.

- **Product importance:** The company has categorized the product in the various parameter such as low, medium, high.
- **Gender:** Male and Female.
- **Discount offered:** Discount offered on that specific product.
- **Weight in gms:** It is the weight in grams.
- **Reached on time:** It is the target variable, where 1 Indicates that the product has NOT reached on time and 0 indicates it has reached on time.

3. Methods

i. K-Means Clustering Algorithm

When you have unlabeled data, you can utilize K-means clustering as a sort of unsupervised learning. This algorithm's purpose is to locate groups in the data, with K representing the number of groups. Based on the features provided, the algorithm iterates to assign each data point to one of K groups. Feature similarity is used to group data points.

ii. Agglomerative Clustering Algorithm

The most frequent type of hierarchical clustering used to put objects in clusters based on their similarity is agglomerative clustering. AGNES is another name for it (Agglomerative Nesting). Each item is first treated as a singleton cluster by the algorithm. Following that, pairs of clusters are merged one by one until all clusters have been merged into a single large cluster holding all items.The output is a dendrogram, which is a tree-based representation of the objects.

iii. DBSCAN Clustering Algorithm

DBSCAN (density-based spatial clustering of applications with noise) is a widely used data clustering algorithm in data mining and machine learning. DBSCAN combines together points that are close to one other based on a distance measurement and a minimum number of points based on a set of points. It also identifies sites in low-density areas as outliers.

iv. MeanShift Clustering Algorithm

The Mean Shift clustering algorithm is an iterative method that works on the means of centroids. It is good at finding globular sections in a dataset. It's generally used in computer vision, like reducing an image to a palette of colors. Mean Shift is not very scalable due to the fact that it has a complexity of $O(T*n*\log(n))$ for smaller dimension counts and $O(T*n^2)$ for larger dimensions in scikit, where T is the number of points and n is the number of samples.

v. Birch Clustering Algorithm

The BIRCH algorithm, which stands for "balanced iterative reducing and clustering using hierarchies", is an incremental, memory efficient, unsupervised hierarchical clustering algorithm. It generally requires one pass over the dataset, because of that it's quite suitable for large datasets that might not fit into memory. It's similar to the Mini Batch K-Means algorithm. For the inputted data, the BIRCH algorithm creates a tree where the centroids of the clusters (among other data) are stored on the leaf nodes. Outliers aren't a big issue for the algorithm, because they are generally removed in an early processing stage. For big dimensions Mini Batch K-Means might perform better than BIRCH. But if the desired output has a large number of subclusters, BIRCH should be preferred.

vi. Mini Batch K-Means

The Mini Batch K-Means is a modified version of the K-Means algorithm that uses mini-batches, which are randomly selected parts of the inputted data. These batches help to reduce the computation cost while outputting slightly better results than K-Means for most datasets. The algorithm can be explained in two steps. First step is where we randomly take samples from the dataset and attach them to the closest centroid. In the second step we update the centroids for each small sample we have put in the mini batches. The two steps are repeated until we converge.

Mini batching runs quicker than the original K-Means algorithm but the output might be slightly worse in some cases. This wasn't true for our datasets, the difference between the two outputs were negligible and mini batching reduced the computation time by a small amount, but because our datasets aren't very large, they both run relatively fast.

vii. Hierarchical Dendrogram

Hierarchical clustering is where you build a tree from the given data points and trying to produce meaningful clusters between every data point. In other words after clustering execution resulted graph looks like round-robin tournament scheduled graph. This algorithm begin by giving each data point its own cluster. After each phase, the most comparable similar two clusters are combined into a single new cluster which ideally ends up as a meaningful represented classification schema. There are 3 linkage type implemented in this paper. Complete, single and average linkage. In complete linkage, points will be merged in each step to the two clusters whose merger has the smallest diameter. In single linkage, points will be merged in each step to the two clusters whose two closest members have the smallest distance. In average linkage, points will be merged in each iteration to the pair of clusters with the highest cohesion.

viii. OPTICS Clustering Algorithm

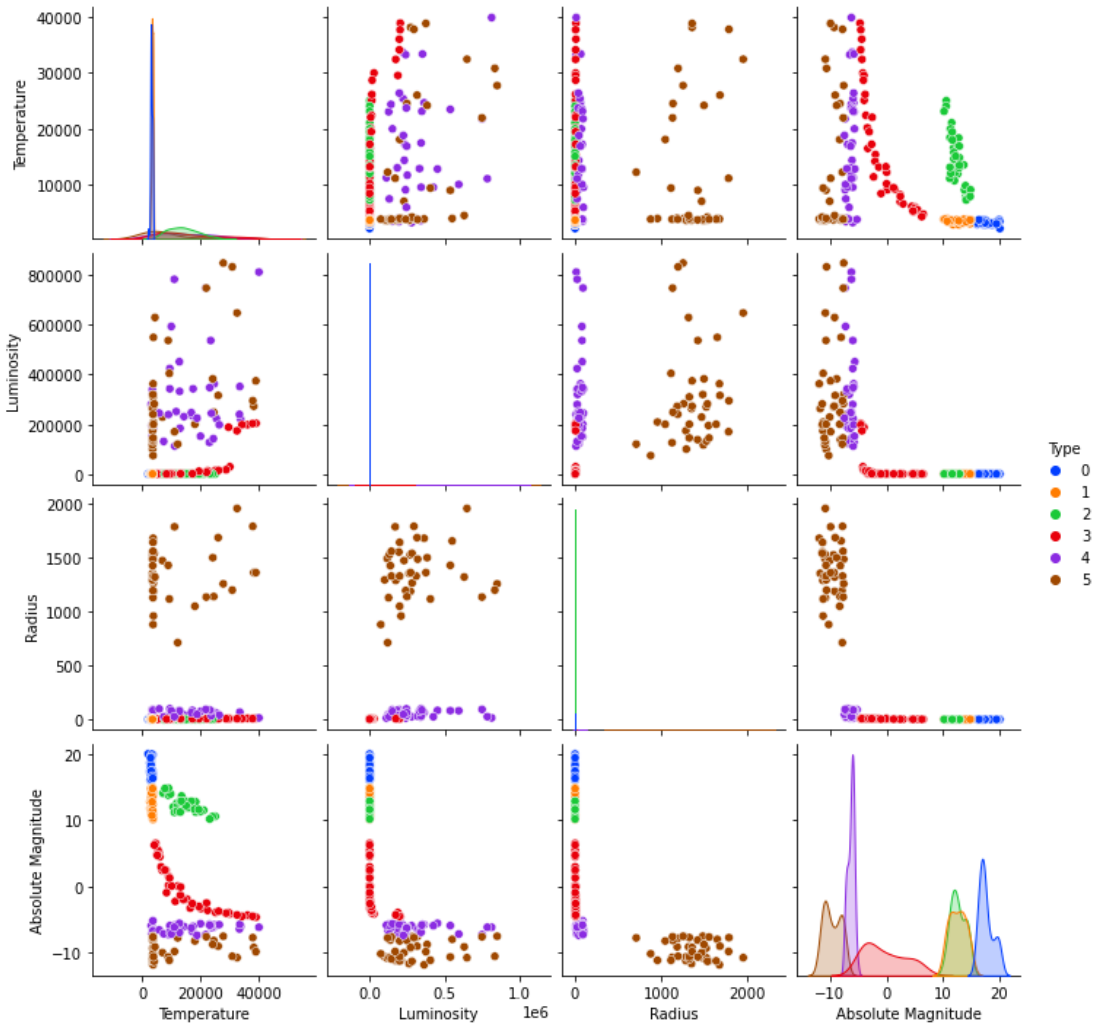
OPTICS Clustering stands for Ordering Points to Identify Cluster Structure. This clustering algorithm is considered as upgraded version of DBSCAN clustering algorithm. The reason for that it addresses two weaknesses of DBSCAN and executes the algorithm. Core Distance and Reachability Distance. If the given point of data is not a Core point, then Core Distance is undefined. This algorithm does not deliberately partition the data into groups. Instead, it creates a visualization of reachability distances and clusters the data using this visualization.

ix. Affinity Propagation Algorithm

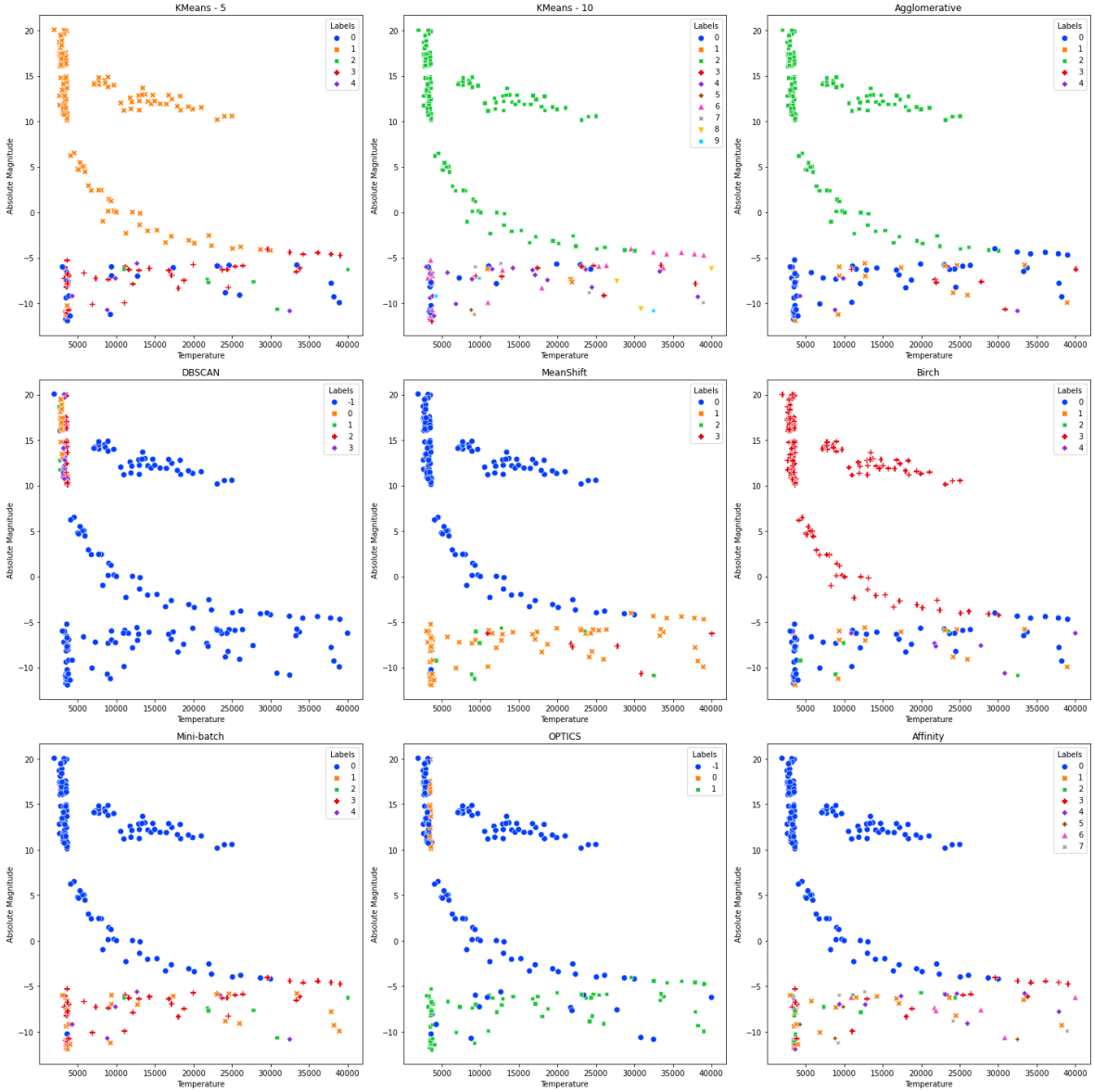
Affinity Propagation is a clustering algorithm unlike others. This algorithm does not take into account the number of clusters before execution. This algorithm based on message-passing procedures. This procedure brings us one keyword which is exemplar. Once one data point is associated with one of its target data point, that target becomes the point’s exemplar. All points with the same exemplar are placed in the same cluster. Affinity Propagation chooses the number of clusters based on the data provided.

4. Experimental Results

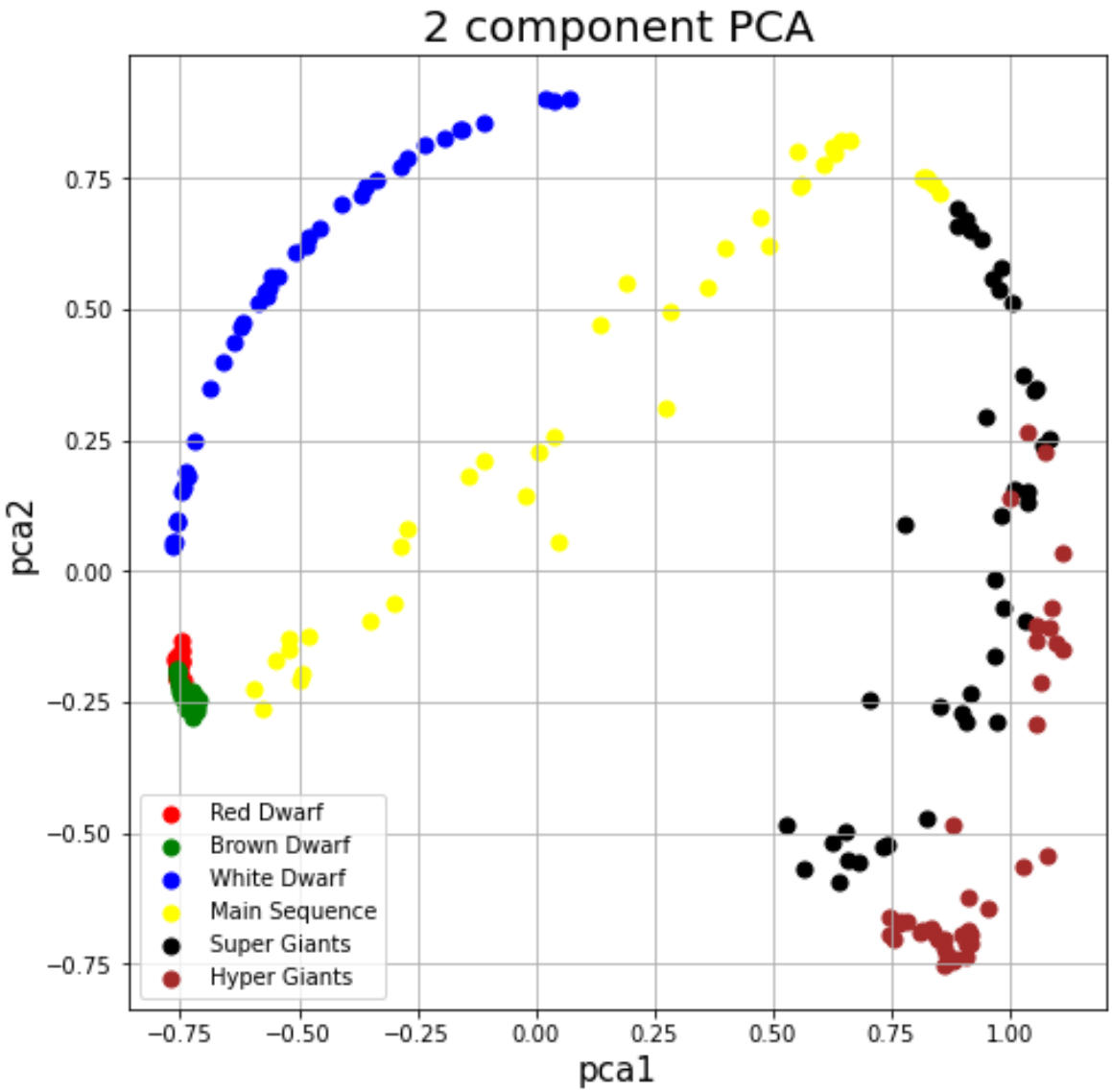
For the star classification dataset we have matrix comparison for the given data of the inputs.



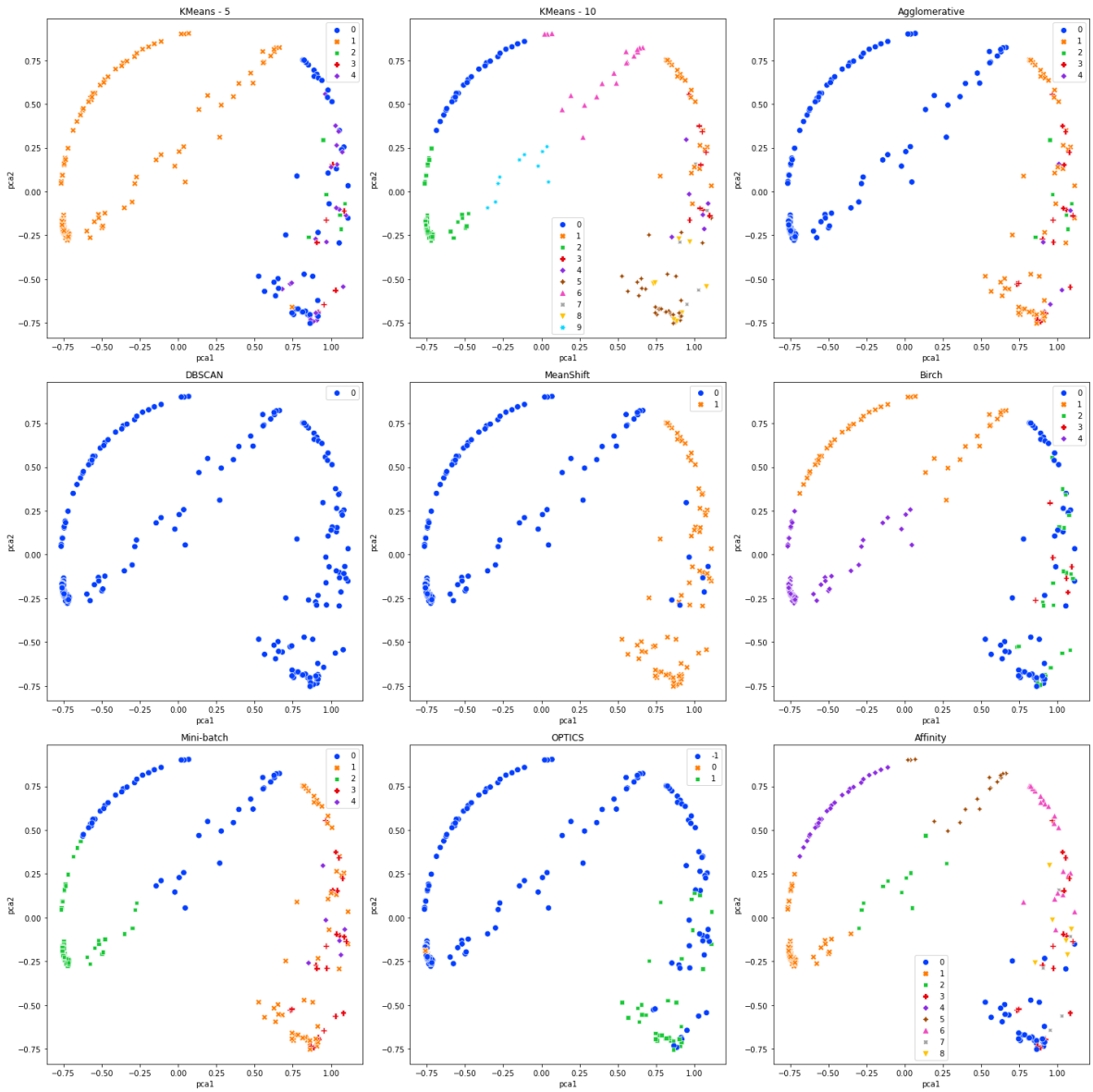
Below is an implementation and comparison of clustering algorithms for star classification dataset. Essentially 6 type given on the dataset for their classification but clustering algorithms intend to find 5 different clusters.



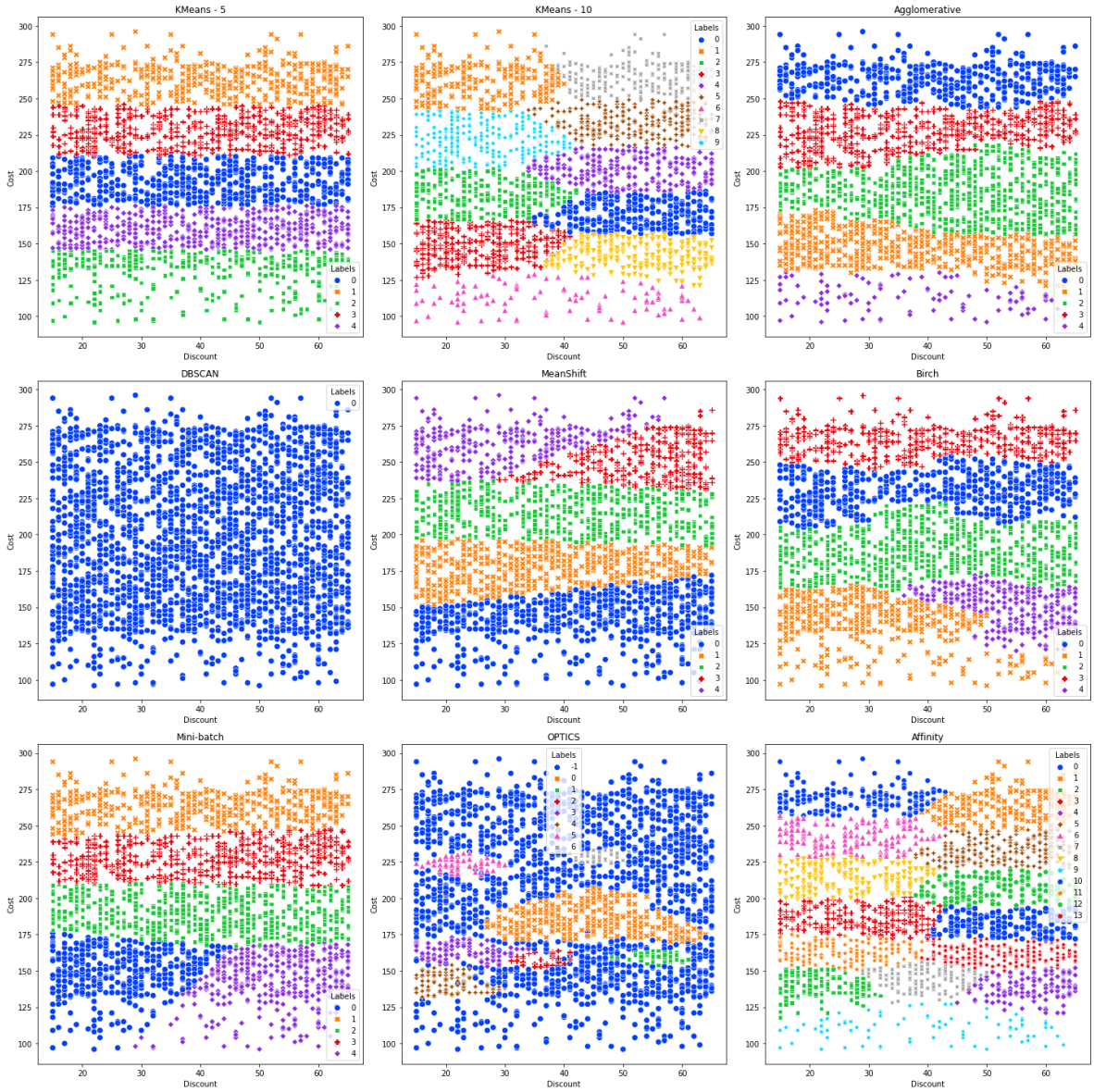
Star classification dataset essentially has 4 different content for classification which we can think as 4-dimension dataset. After applying PCA analysis on the given data has become 2-dimension for classification. After PCA we are normalizing the data so that the data approximately follows a Gaussian distribution. And we have much better results for the classification with this pre-processed data.



Below is an implementation and comparison of clustering algorithms for star classification dataset but with couple more steps. We mentioned and showed visualization of clustering algorithm comparison of star classification dataset. This comparison made after in terms of PCA application and normalization steps.



Below is an implementation and comparison of clustering algoritms for e-commerce shipping dataset. We made this comparison in order to see what happens if data points covered the all coordinate system.



Without applying different clustering algorithms, lazypredict library from python shows which classifiers is best for classification accuracy and balanced accuracy. Since we are dealing with small dataset in this classifier comparison most of the classifiers ended up almost 100% accuracy.

XGBClassifier	1.00	1.00
RandomForestClassifier	1.00	1.00
Perceptron	1.00	1.00
ExtraTreesClassifier	1.00	1.00
GaussianNB	1.00	1.00
LinearDiscriminantAnalysis	1.00	1.00
SVC	0.98	0.98
SGDClassifier	0.98	0.98
KNeighborsClassifier	0.98	0.98
LabelPropagation	0.98	0.98
LabelSpreading	0.98	0.98
LinearSVC	0.98	0.98
LogisticRegression	0.98	0.98
PassiveAggressiveClassifier	0.98	0.98
LGBMClassifier	0.98	0.98
BaggingClassifier	0.98	0.98
DecisionTreeClassifier	0.98	0.98
ExtraTreeClassifier	0.96	0.97
NuSVC	0.96	0.97
CalibratedClassifierCV	0.96	0.97
NearestCentroid	0.90	0.90
RidgeClassifier	0.83	0.85
RidgeClassifierCV	0.83	0.85
BernoulliNB	0.81	0.83
AdaBoostClassifier	0.58	0.67
QuadraticDiscriminantAnalysis	0.17	0.21
DummyClassifier	0.17	0.16

5. Conclusions

After evaluations we observed that on small dataset, clustering algorithms intend to cluster big chunks first, then clusters the crowded small chunks. This approach from clustering algorithms is bad for one reasons which is results will be different than the real labeled data visualization, if correlation between two variables does not strong and one of the variable does not solid feature for classification.

On large dataset, clustering algorithms intend to find clusters around one variable which is not sufficient to observe meaningful results from dataset. But on the other hand if all coordinate system covered by given data, these clustering algorithms could produce meaningful visualization for specific related variable in the coordinate system.

This paper presents a simple and efficient way for assigning different clustering mechanism for data points to see them in different clusters. We can safely say that these clustering algorithms worked well on both datasets on generally.

6. References

1) https://www.kaggle.com/brsdincer/star-type-classification
2) https://www.kaggle.com/prachi13/customer-analytics
3) Gupta, Manoj & Chandra, Pravin. (2019). A Comparative Study of Clustering Algorithms.
4) Jain, A.K., Murty, M.N. and Flynn, P.J. (1999) 'Data clustering: a review' ACM Comput. Surv. 31, 3, 60 pages.
5) J. Oyelade et al., "Data Clustering: Algorithms and Its Applications," 2019 19th International Conference on Computational Science and Its Applications (ICCSA), 2019, pp. 71-81, doi: 10.1109/ICCSA.2019.000-1.
6) K. bindra and A. mishra, "A Detailed Study of Clustering Algorithms," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017, pp. 752-757, doi: 10.1109/CTCEEC.2017.8454973.
7) J. Khalfallah and J. Ben Hadj Slama, "A Comparative Study of the Various Clustering Algorithms in E-Learning Systems Using Weka Tools," 2018 JCCO Joint International Conference on ICT in Education and Training, International Conference on Computing in Arabic, and International Conference on Geocomputing (JCCO: TICET-ICCA-GECO), 2018, pp. 1-7, doi: 10.1109/ICCA-TICET.2018.8726188.
8) https://en.wikipedia.org/wiki/Asteroid_spectral_types
9) https://scikit-learn.org/stable/modules/clustering.html

APPENDIX

1- One of the clustering algorithms visualization (Agglomerative Clustering)

```
from sklearn.cluster import AgglomerativeClustering
agglom = AgglomerativeClustering(n_clusters=5, linkage='average').fit(X)
X['Labels'] = agglom.labels_
plt.figure(figsize=(12, 8))
sns.scatterplot(X['Temperature'], X['Absolute Magnitude'], hue=X['Labels'],
                palette=sns.color_palette('bright', 5))
plt.title('Agglomerative with ' +
          str(np.unique(agglom.labels_).shape[0]) + ' Clusters')
plt.show()
```

2- Github repository of project

https://github.com/musacam/clustering_algorithms_ml