Comparative Analysis of Different Clustering and Classification Algorithms.

Prathamesh Kulkarni (Student ID: 36004026)
Dept. of Computer Science and Communication
Lancaster University
Lancaster, England
p.p.kulkarni@lancaster.ac.uk

***Abstract*—The paper presents a range of clustering techniques like K-Means, Agglomerative Clustering, DBSCAN, and classification algorithms like logistic regression, support vector machines, and gradient boosting algorithms. These techniques were applied to two different datasets, while clustering was used for one dataset and classification for the other. Each model was then assessed for performance.**

***Index Terms*—Clustering, Classification, K-Means Clustering, Agglomerative Clustering, Logistic Regression, Support Vector Machines, Gradient Boosting Algorithms**

I. Introduction

A critical aspect of learning machine learning is understanding clustering and classification tasks. In this paper, multiple clustering and classification algorithms are implemented along with a discussion of how they work and when to use them. In the clustering section, techniques such as K-Means, Agglomerative Clustering, and DBSCAN are discussed, as well as aspects such as how to use the elbow curve to identify the right number of clusters to create or how the silhouette score is used to test the performance of the clustering model. This paper explains whether PCA and standardized data applied to the same models could lead to more interpretable clusters and better cluster quality. Classification algorithms such as Logistic Regression, Support Vector Machines, and Gradient Boosting are studied. Also discussed are a variety of evaluation metrics to verify that the models are performing as expected.

II. Pre-processing

In order to make sure our models are provided with quality data, it is very important to preprocess the data before applying models to the datasets. Using preprocessed data ensures better results. The first data set was checked for null values, and none were found. An outlier detection technique using the z-score was used to remove any outliers since algorithms like K-Means are affected by outliers. The correlation matrix was used to select and remove features. Next, the data was standardised. Following standardization, the standardized data were then fed into the PCA algorithm, resulting in a new data set consisting of 4 principal components. In the second dataset, no null values were found. To balance the dataset, first 16 points were removed. As a good practice, the data was shuffled. The labels were replaced with new ones, i.e. class 1 labels were given a label value of 0, class 2 labels a label value of 1. This makes it easier to work with libraries that only accept 0 and 1 for labels. 80 percent of the data were split into training data and 20 percent into testing data. The preprocessing process ended at this point.

III. Clustering

Clustering involves grouping together similar pieces of data. It is, in other words, the assignment of similar clusters to similar examples of data or groups of similar examples. The clustering is unsupervised, i.e., no labels are available during the training. When labels are available, it becomes a classification task. Each point is clustered into its own class/label/group. A classification model can be trained by using acquired labels, or can simply be used to identify patterns in data and analyze different groups.

It is possible to use clustering as an intermediate step in other data mining problems. For instance, outlier detection or identifying labels for classification could be carried out with clustering. Filtering and segmenting similar customers or users can be performed using it. It is useful for marketing, targeting and recommending products and services to customers/users. Data clustering can be used to summarize a large dataset, for example by showing the different clusters of data to increase the interpretability of the data. Using clustering, the data can be analyzed to determine the most significant events and trends. Images, audio files, and videos can be categorized according to their similarities. Each of these applications constitutes a large application area for clustering.

Understanding clustering's applications helps us choose the appropriate clustering techniques based on our data and desired results. In density-based clustering, clusters are developed based on the number of data points. The region with the most data points is considered a cluster. Hierarchical clustering looks at the data points as clusters, which are then put into their clusters based on distance metrics. As a result, data points which are close to each other are considered clusters. Unlike other clustering techniques, in fuzzy clustering, where a single data point is confirmed to belong to only one cluster, fuzzy clustering provides the probability of a single data point belonging to each cluster. Partition clustering involves the creation of a set of clusters, which is then assigned to data points, with an iterative process based on distance, reassigning data points between clusters. A cluster is assigned to the

data point that is closest to its centroid. Based on grid-based clustering, the data is represented as grids. The data points themselves are less important with this method than the value space surrounding them. It computes the density of the cells after segmenting the data into cells. This is used to identify clusters. [2]

For the analysis of our data, we used two algorithms, which are K-means and agglomerative.

K-Means Clustering

The K-means algorithm is a partitional clustering algorithm. It is one of the most common algorithms for clustering data. There are K clusters in total. The model can then determine which cluster new data belong to after the clusters have been generated. The algorithm assigns the data points to their clusters based on similarities between them.

Now let's look at how the algorithm works. Clustering is accomplished by using distance measures to find similarity between data points as part of the k-means algorithm. The default distance measure is Euclidean. To ensure k-means isn't affected by outliers, we need to eliminate them first.

1) Our first step is to select k, the number of clusters.
2) The data points are randomly selected for clustering into k points.
3) It is possible to choose a distance measurement, such as euclidean distance, whose formula is:

$$Distance = \sqrt{(Y_2 - Y_1)^2 + (X_2 - X_1)^2}$$

   We calculate the distance from each point in the data space to each cluster centre.
4) There are estimated 1800 points of data and if we choose k as 3, then we need to calculate approximately 5400 distances.
5) Data points in the data space are assigned a cluster centre based on their location.
6) It is then necessary to verify that the cluster centers currently picked at random are, in fact, the centers of gravity of each cluster.
7) Using the mean of the data points in each of the k clusters, cluster centers are calculated.
8) For each data point, there is a need to recalculate the cluster assignment if the cluster centers have moved.
9) Thus, repeat step 3 again. The process is complete if none of the cluster centers have shifted.
10) Furthermore, stop condition is defined, since the k means algorithm may take a long time to converge; therefore, we define the stop condition to stop the calculation.

[3] [4]

K-Means is implemented in 3 steps. The elbow method has been used to determine the optimal cluster or value of k that will result in nice clusters. This method selects the best value from a range of possible values. This function calculates the sum of squares of points and the average distance between them. When k increases, the sum of squares within the cluster decreases. As a final step, K value is obtained by plotting the graph between k-values and within-cluster sums of squares. The graph will be examined carefully. It will decline abruptly at some point. The value of k at that point will be considered. As a result of our implementation, k=3.

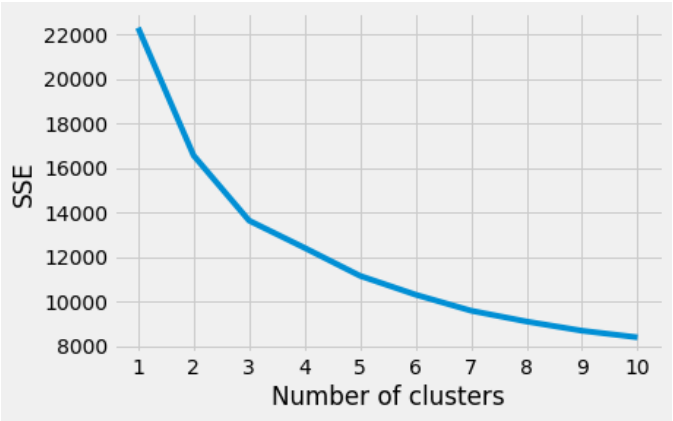

Fig. 1. The plot represents the elbow curve used to find the optimal k value which is the number of clusters, it can be seen that k=3 is chosen from the above graph.

Kmeans library is then used to build the Kmeans clustering model with k=3 found in the previous step. The model is then fitted to our dataset. It then predicts the labels for the data.

Our next step is to visualize the clusters and see if the clusters are nicely separated.

Agglomerative Clustering

Agglomerative hierarchical clustering, is a form of bottom-up clustering.A cluster is defined for each point of data in this clustering method. The algorithm merges cluster pairs as it goes up the hierarchy.

By using a particular proximity measure, a dissimilarity matrix can be constructed to make a dendrogram. Clusters that are closest to each other at each level are merged and the dissimilarity matrix is updated accordingly. The aggregation process continues until all data objects are merged into the final maximal cluster. Our dendrogram would reach its apex at this point, marking the end of the merging process.

Single Linkage: The closest pair is compared to determine similarity. Even if groups have overall dissimilarity, close pairs can merge earlier than is optimal. Complete Linkage: Calculates similarity between pairs that are farthest apart. Outliers can make this merge less-than-optimal if they arise. Average Linkage, or group linkage: Groups of objects are compared instead of individual objects to determine similarity. Centroid Method: Clusters with similar centroid values are merged in iterations. Across all points, the centroid represents the average. Ward's Method (ANOVA based): Every time a new cluster is created, its variance is measured using an index called E (also known as the sum of squares index).

In our implementations, ward criteria is used. When agglomerative clustering is being performed, Ward's criterion is used to calculate the distance between two groups. Agglomeration is also known as Ward's agglomeration, which is the process of merging clusters based on Ward's criteria. In

order to determine the distance, it uses the K-means squared error criterion. In Ward's criterion, which is calculated for any two clusters, Ca and Cb, the increase in the SSE criterion is calculated for the clustering obtained by joining them together. Following is a definition of Ward's criterion [1]:

$$W(C_{a\cup b}, c_{a\cup b}) - W(C, c) = \frac{N_a N_b}{N_a + N_b} d(c_a, c_b)$$

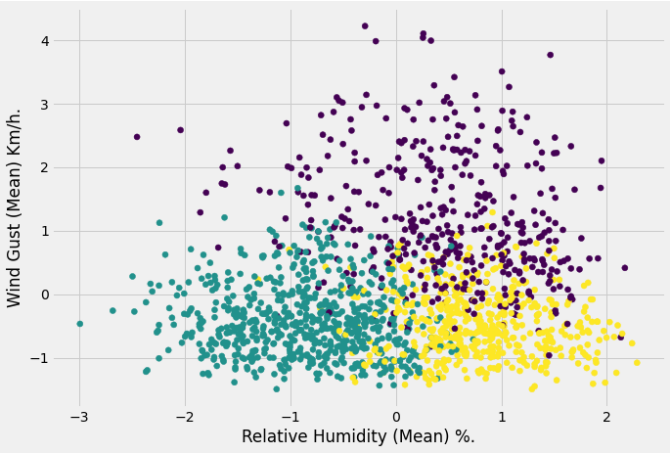

Fig. 2. The above figure represents the clusters that were predicted by using Agglomerative Clustering

DBSCAN

The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering technique is used in machine learning for the separation of clusters containing high and low densities. It does a great job of finding data areas with a high density of observations, as opposed to areas that are not very dense with observations, since DBSCAN is a density-based clustering algorithm. A strong advantage of DBSCAN is its ability to sort data into clusters of various shapes. The key parameters of DBSCAN are as follows:
In order to explore the neighborhood of a point, we will use the epsilon number, which represents the radius of the circle around the point.The minpt metric is used to find the least number of points around a point. The data points can be classified into three categories based on these two parameters: core points, boundary points, and noise points. In this case, let's talk about two points. If two points are neighbors, a density edge joins them. [13]

The DBSCAN process consists of the following steps:

1) Sort the points into categories.
2) Get rid of noise
3) Clusters should have a core point assigned to them.
4) The density points that are connected to the core points should be colored.
5) Color boundary points according to the nearest core point.

Evaluation

Clusters are evaluated by the silhouette score. Every sample of each cluster is assigned the silhouette score. It is necessary to find the following distances between each observation belonging to each cluster in order to calculate the Silhouette score: This is the average distance between an observation and all other data points in the same cluster. This is also known as the mean intracluster distance. It is represented by a. It is the average distance between the observation and all the other data points in the cluster next to it. We can also call this the nearest-cluster distance. We denote the mean distance by b.
Silhouette score, S, for each sample is calculated using the following formula: $S = \frac{(b-a)}{max(a,b)}$ Silhouette scores range from -1 to 1. The cluster is well separated and dense if the score is 1. Resulting values near 0 represent clusters with samples that are very close to their neighbors' decision boundaries. When the samples have been assigned to the wrong clusters, the score [-1, 0] is negative.
Based on both clustering algorithms, the silhouette score for standard data is 0.24 and for PCA data it is 0.32.

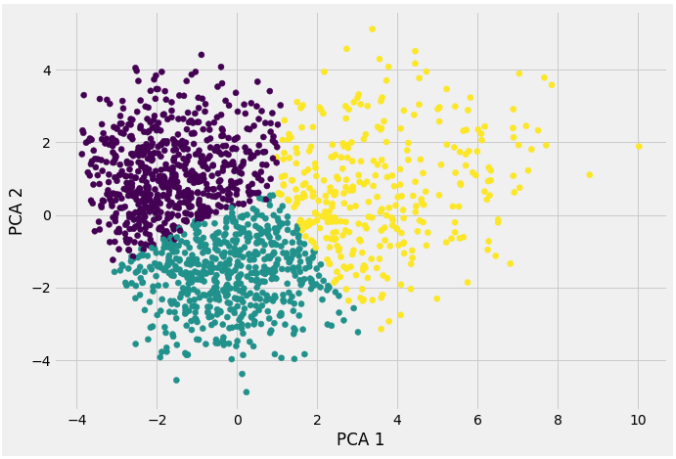

Fig. 3. The above figure shows the clusters that were predicted by the K-Means clustering alogrithm on the PCA data. The clusters appear to be properly separated with no overlap.

IV. Classification

Machine learning and statistics typically involve supervised learning through classification. In this approach, the machine makes new observations or classifications based on the data it is given. To categorize things, data is grouped into similar groups. Targets, labels, and categories are common terms used to describe the types of data points. The main objective is to identify the class into which the newly collected data will be assigned. A set of feature variables is used to learn the relationship between a target variable and a set of feature variables. A broad range of practical problems can be expressed as associations between feature and target variables, making this model useful for a wide variety of problems.

The classification types can be classified into four broad categories. Binary classification is a classification that categorizes an output variable into two groups, such as yes or no, existing or not existing. It is possible to categorize data in more than two groups when using multiclass classification. In multilabel classification, multiple labels can be assigned to

the same instance. In cases of imbalanced classification, the frequency of one class of input variables is higher than the other classes.

Several methods can be used to apply classification, but we chose logistic regression, support vector machine, and gradient boosting.

Logistic Regression

It is possible to determine the likelihood of class by using logistic regression or linear regression. It is suitable to use logistic regression when the dependent variable is categorical or binary. In contrast, logistic regression is not appropriate to predict continuous variables like age, size, and so on. In any event, logistic regression refers to the association between a binary dependent variable and a variable or variables of varying levels of order, ordinal, or ratio. Models of logistic regression depend on a fixed number of parameters, which change depending on the number of input features, and give categorical predictions, such as identifying plants that belong to a certain species.

Unlike linear regression, logistic regression does not fit straight lines directly to the data. Rather than fitting a straight line to our observations, S curve is fitted, called a sigmoid. A curve of this shape can map any real-valued number into a value between 0 and 1, but never exactly at those limits. sigmoid function is given by $sigmoid(z) = \frac{1}{1+e^(-z)}$

In order to make a prediction, first compute a weighted sum and then enter this into the sigmoid function:

1)Calculate weighted sum of inputs z = w0 + w1.x1 + ...wn.xn

2)Calculate the probability: $probability(z) = \frac{1}{1+e^(-z)}$

The S-shaped line can be fitted to the data in several ways to train a Logistic Regression model. Calculating the parameters of the model (the weights) can be done through iterative optimisation algorithms such as Gradient Descent or by using probabilistic methods such as Maximum Likelihood.

Finally, determine a threshold. Assign a class depending on whether the probability is above or below the threshold. [6]

Support Vector Machines

In support vector machine algorithms, a hyperplane is found in an N-dimensional space (N equals the number of features) that identifies the data points.Linear conditions are used with SVM methods in order to separate the classes. Ideally, the two classes should be separated as much as possible by a linear condition. With a carefully crafted multivariate split condition, SVM classifiers can be viewed as single-level decision trees. Given that the single separating hyperplane determines the effectiveness of the approach, it is crucial to define that separation carefully. The maximum margin hyperplane is by far the most important criterion for SVM classification. It is possible to choose many possible hyperplanes to separate the two classes of data points. The plane is determined by its maximum margin, that is, the distance between two independent classes of data points. Future data points can be classified with greater confidence when the margin distance is maximized Hyperplanes are used to help classify data points. Based on the data points lying on either side of the hyperplane, data points can be classified differently. Additionally, the dimensions of hyperplanes depend on the number of features. Whenever the input contains only two features, the hyperplane is a line. Three features in the input can result in a two-dimensional hyperplane. This becomes more difficult as the number of features increases. Based on their proximity to the hyperplane, support vectors indicate how an object should be positioned and oriented with respect to the hyperplane. These support vectors maximize the margin of the classifier. When the support vectors are deleted, the position of the hyperplane will be altered. SVMs are built using this method. The SVM process takes the linear function's output, and if it's greater than 1, it is identified with one class. If it's lower than 1, it's identified with another class. When we change the threshold values in SVM to 1 and -1, we obtain this reinforcement range of values ([-1,1]) that is known as the margin. [5], [7]

Gradient Boosting Machines

It is a distinct approach from bagging to correct errors for each new tree in boosting. A final model aggregates all of the individual trees and builds new trees from scratch using bootstrapping. When boosting, a new tree is created based on the previous tree. It is not possible for trees to function independently; they are interconnected. In gradient boosting, corrections are made exclusively based on the errors from the previous tree's predictions. This means that each new tree is adjusted entirely based on the previous tree's errors. The gradient boosting method looks for mistakes in a tree and builds a new tree around those mistakes. This tree doesn't consider predictions that are already correct.

The next step is to understand how gradient boosting works.

1) Apply the decision tree to the data
2) Rather than predicting from a test set, gradient boosting predicts from a training set. During the training phase, we do this in order to calculate the residuals as we need to compare predictions. After all trees have been built, testing of the model occurs.
3) The residuals represent the difference between the predictions and the target column.
4) Fit the new tree on the residuals
5) Repetition of steps 2-4 should result in the residuals approaching 0 as they move towards either a positive or negative direction. If necessary, this process can take place for many trees.
6) Make predictions for each tree with the test set to sum the results

Gradient boosting has been applied in our implementation using Light GBM [8]–[10]

Evaluation of Classification Models

Accuracy: A model's accuracy is determined by the number of correctly classified data over its total number of classifications.

$Accuracy = \frac{Numberofcorrectpredictions}{Totalnumberofprediction}$

Accuracies for each models are:

1) Logistic Regression: 0.97
2) Support Vector Machine: 0.97
3) Gradient Boosting: 0.97

Precision: How many of the positive identifications were correct

$$Precision = \frac{TruePositives}{TruePositives+FalsePositives}$$

1) Logistic Regression: 0.94
2) Support Vector Machine: 1
3) Gradient Boosting: 0.94

Recall: How many of the actual positives were correctly identified

$$Recall = \frac{TruePositives}{TruePositives+FalseNegative}$$

1) Logistic Regression: 1
2) Support Vector Machine: 0.94
3) Gradient Boosting: 1

F1-score: A model's F-score, also called an F1-score, measures the accuracy of that model over a dataset.

$$F1-score = \frac{TruePositives}{TruePositives+\frac{1}{2}(FalsePositive+FalseNegative)}$$

1) Logistic Regression: 0.97
2) Support Vector Machine: 0.97
3) Gradient Boosting: 0.97

ROC curve: This is a graph that displays how the diagnostic ability of a binary classifier system varies with its discrimination threshold.

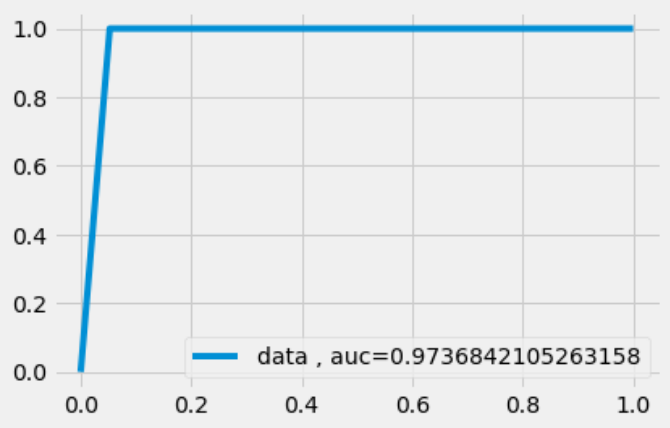

Fig. 4. The figure represents the ROC curve obtained from the Gradient Boosting algorithm

Confusion Matrix: This is a table that shows how classifiers perform on test data whose true values are known. It displays the values of the true positives, true negatives, false positives and false negative in a tabular format.

V. Conclusion

Various clustering algorithms and classification methods have been successfully implemented. Two data sets were analyzed using the k-means and agglomerative algorithms, one with original features, the other with components of PCA. Both algorithms were tested using both data sets in order to see the effects they had on cluster interpretation. In both data sets, there was no significant difference between the silhouette scores of the two algorithms. Nevertheless, the data set with PCA components was able to portray clusters more accurately than the original data set. The clusters were not clearly distinguished for most of the features. It could be that the features were not related at all to form clusters, but at least now we know which features to use to determine relationships during analysis. Alternatively, the models may not have been able to properly cluster the data, as evidenced by the relatively low silhouette scores. Our analysis reveals that all classification algorithms provide a high level of accuracy of 97 percent. Moreover, the confusion matrix shows that the classes were correctly classified, demonstrating that there was no overfitting. In contrast, higher accuracy of a model should always be treated as a matter of caution as high accuracy does not always guarantee that the model will perform well for new or large data sets. Due to the fact that the models are trained on a very small amount of data.

Acknowledgment

For the lectures and the lab sessions, I would like to acknowledge the material provided by professor Angelov, Plamen P. Furthermore, I would like to acknowledge the creators of libraries such as numpy, pandas, seaborn, sklearn, svm, lightgbm, and matplotlib. Additionally, I want to acknowledge Abdul Ghani Khan's help with this report.

References

[1] Charu C. Aggarwal, Chandan K. Reddy, "Data Clustering" Chapman and Hall/CRC", August 2013.
[2] Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", O'Reilly Media, Inc., September 2019.
[3] Dr. Basant Agarwal, Benjamin Baka, "Hands-On Data Structures and Algorithms with Python", Packt Publishing, October 2018.
[4] Imran Ahmad, "40 Algorithms Every Programmer Should Know, Packt Publishing", June 2020.
[5] Charu C. Aggarwal, "Data Classification, Chapman and Hall/CRC", July 2014.
[6] David W. Hosmer Jr., Stanley Lemeshow, Rodney X. Sturdivant, "Applied Logistic Regression, 3rd Edition", Wiley, April 2013.
[7] Aurélien Géron, "Understanding support vector machines", O'Reilly Media, Inc., April 2017.
[8] Giuseppe Ciaburro, Prateek Joshi, "Python Machine Learning Cookbook - Second Edition", Packt Publishing, March 2019.
[9] Stephen Klosterman, "Data Science Projects with Python - Second Edition", Packt Publishing, July 2021.
[10] Corey Wade, Kevin Glynn, "Hands-On Gradient Boosting with XGBoost and scikit-learn", Packt Publishing, October 2020.
[11] S. Na, L. Xumin and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, 2010, pp. 63-67, doi: 10.1109/IITSI.2010.74.
[12] Angelov, Plamen P., "Autonomous learning systems: from data streams to knowledge in real-time", John Wiley and Sons Ltd, 2012
[13] Giuseppe Bonaccorso, "Hands-On Unsupervised Learning with Python", Packt Publishing, February 2019.

APPENDIX

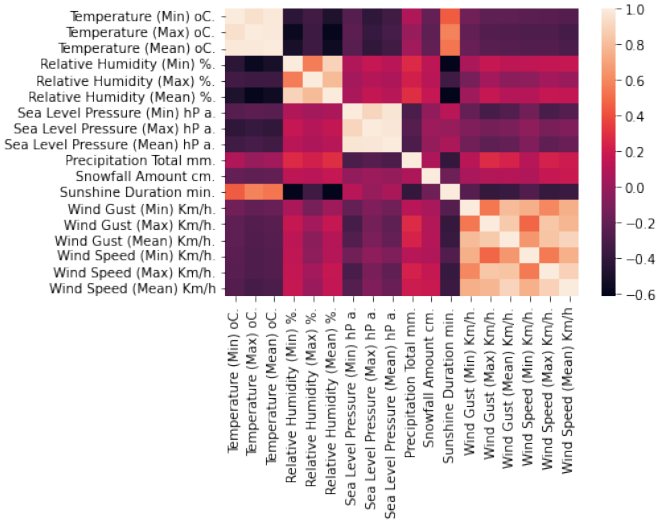

Fig. 5. Heatmap showing the correlation between features

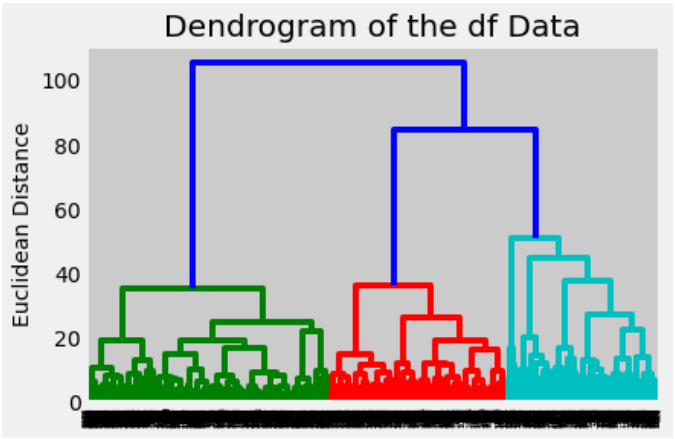

Fig. 6. The figure shows the dendrogram

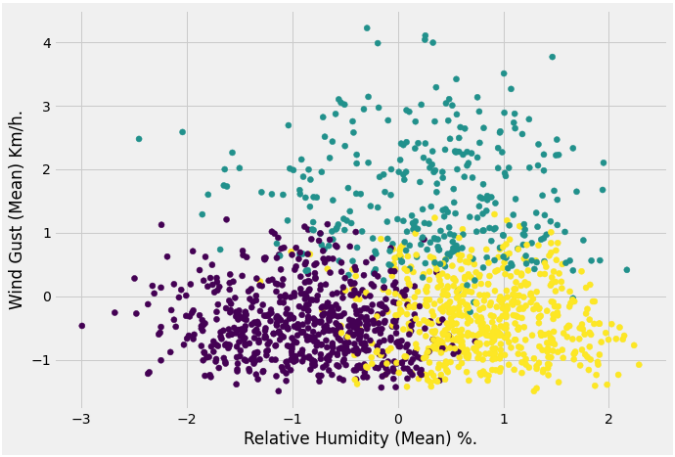

Fig. 7. K-Means clustering for normal data

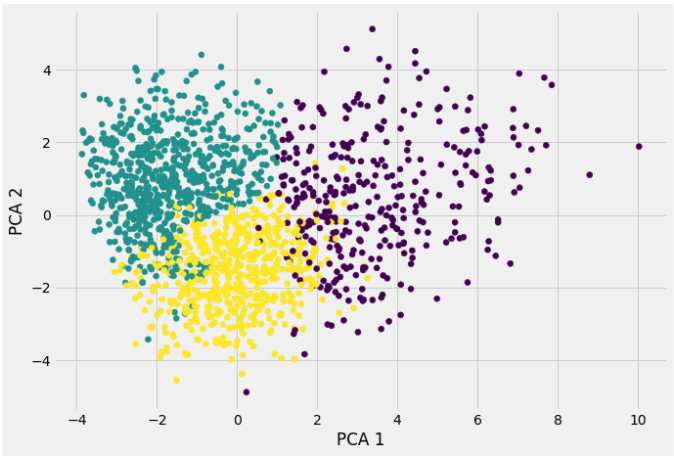

Fig. 8. Agglomerative clustering for pca data

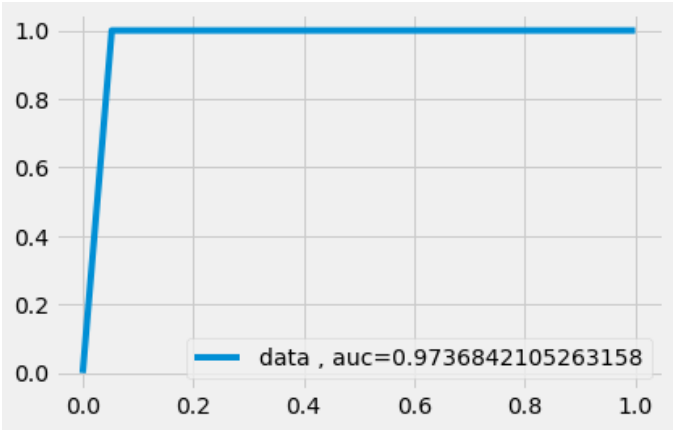

Fig. 9. ROC curve of Logistic Regression Model

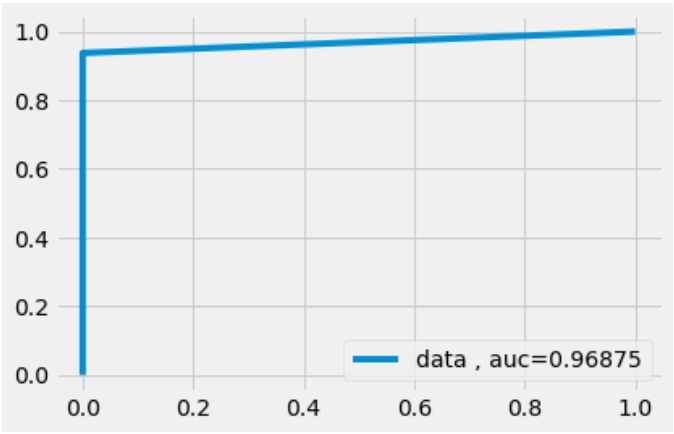

Fig. 10. ROC curve of Support Vector Machine model

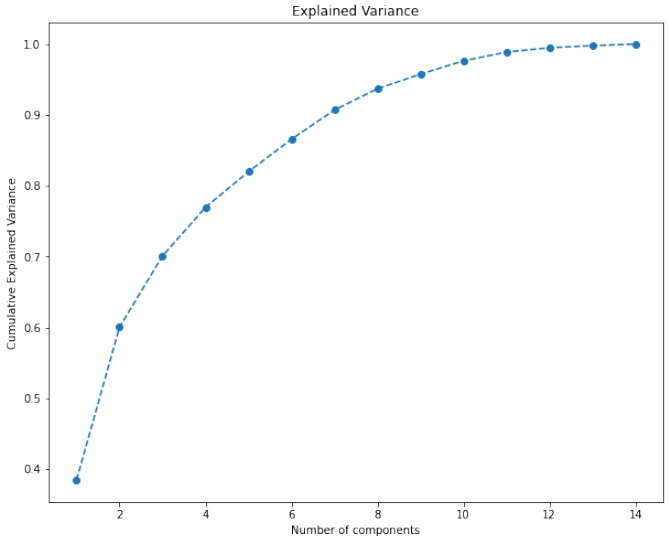

Fig. 11. Variance plot used to identify the number of PCA components to be taken