Anthony Gray
EN.605.649.81
Intro to Machine Learning
Project 2

Abstract

K-means and Stepwise Forward Selection were implemented and run on three datasets. Their performance was measured by silhouette coefficient, and the extracted features used to plot the clusters for each dataset. K-Means performed well, producing consistently high average silhouette coefficients, and the performance of SFS varied by dataset. With better controlled evaluation of SFS and more optimized code for the larger dataset, SFS may have performed as well as K-Means.

Problem Statement and Algorithms

The K-Means algorithm (3) was implemented accepting a dataset and a *k* value as parameters. It initialized *k* random points and repeatedly assigned all points to a cluster, then moved those points to the mean position of that cluster in Euclidean space. This process is repeated until the centroids stop moving and the dataset with updated cluster values is returned. Random centroids were initialized randomly within bounds based on the data. For each attribute, the minimum centroid value was equal to the minimum of the attribute value minus twice its standard deviation. The maximum centroid value was equal to the maximum attribute value plus twice the standard deviation. Each centroid dimension was initialized within this range, calculated for each attribute.

Stepwise Forward Selection (2) was implemented as a wrapper method, accepting a dataset, clustering algorithm, evaluation algorithm, and *k* parameters. It builds a subset of the attributes one at a time by repeatedly adding a new feature to the existing set, clustering the data based on these features and evaluating the clusters. When the clustering performance stops improving, the algorithm returns the feature subset that provided the best score.

Clustering performance was evaluated using the silhouette coefficient (7). For each point in the dataset, the average distance between it and all the other points in each cluster is calculated. The average distance between it and the other points in its cluster is subtracted from the average distance to the points in the nearest cluster, and that difference is normalized by dividing it by the larger of those two values. All such values for each point in the dataset are then averaged, giving a useful measurement of cluster separation, with values close to 1 indicating good separation, while values close to -1 indicating poor separation.

These algorithms implemented in python and were employed to find the most descriptive feature subset for the Iris (4), Glass (5) and Spambase (6) datasets.

Because the silhouette coefficient is being used to evaluate the results of the K-means clustering algorithm, the Stepwise Forward Selection algorithm will find the feature set that maximizes the coefficient. The range of the dataset's attributes may also play a role in SFS selection. If the dataset has a few features with greater ranges, it should be possible to cluster based on those features with more accuracy than with features with lesser ranges. Silhouette Coefficients will generally be close to 1 if K-Means is properly implemented, and the reduced feature sets provided by SFS should plot into observable clusters if they contain two elements.

Procedure/Tuning

Data Sets:

Name	Instances	Attributes	Classes	Average Attribute Range	Greatest Attribute Range
Iris	150	4	3	2.86	5.9 (Petal Length)
Glass	214	9	7	4.06	10.76 (Ca)
Spambase	4601	57	2	475.57	15840 (crl_total)

All three datasets were obtained from UCI Machine Learning Repository (1). Datasets were treated with minimal processing, as no missing values were present. Column names in the spam set were abbreviated for readability, with “word frequency” being shortened to “wf,” “character run length” becoming “crl” and so on. Some data types were altered, but all data instances were treated as points in Euclidean space with dimensionality equal to the number of attributes. The Class feature of all datasets was replaced with a Cluster feature, initialized to -1.

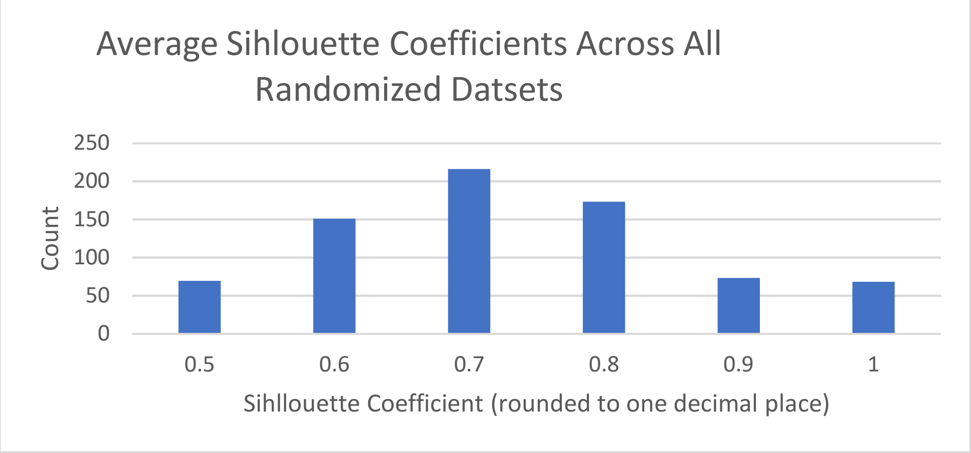
All datasets were clustered by K-Means, using Stepwise forward selection as a wrapper to choose the best features. *K* values were set to the number of classes known to be present in each dataset. The Iris and Glass datasets were run in their entirety, but a single run of the whole Spambase dataset took several hours, so subsequent runs were performed on a random sample of 175 instances from that set, both to reduce the processing time per run and to keep the number of samples similar to those of the other two datasets.

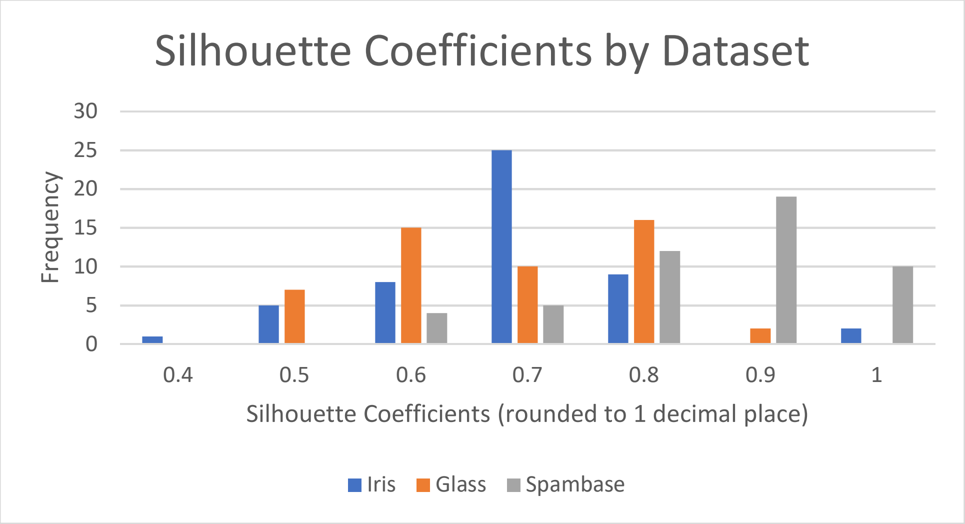
Each dataset was run through the algorithms several times and the reduced features sets and average silhouette coefficient were recorded. Then each dataset was run through the algorithms with their features in a random order to see if the order of the feature set had an impact on the performance of the greedy SFS algorithm. The frequencies of the appearance of each feature were recorded, along with the silhouette coefficients for each run. Each dataset was run with a randomized feature set 250 times, resulting in 750 randomized runs across all datasets. The random SFS algorithm was also run on each dataset 50 times, and the silhouette coefficients were recorded by dataset.

Finally, clusters were plotted with the most returned features for each dataset. Some plots are included in the Summary and Appendix sections.

Results

The silhouette coefficients generated by the K-Means clustering algorithm were consistently positive, with all but one observation, with or without randomized features being greater than 0.5 across all runs. The histogram below shows the average silhouette coefficients collected across 750 randomized feature runs. A histogram of the average silhouette coefficients from 50 additional runs per dataset is also shown.





The unrandomized SFS algorithm consistently returned the features Petal Length and Petal Width from the Iris dataset. Out of 250 random SFS runs, both features were returned the most times.

The unrandomized SFS algorithm returned Ba every time, often paired with Fe when run on the Glass dataset. However, the randomized SFS algorithm returned RI most often, followed by Ba. Fe was the third least likely feature to be returned by the random algorithm.

The unrandomized SFS algorithm returned a feature set of crl_total and crl_longest every time it was run on the Spambase dataset, and crl_total was returned frequently with the randomized SFS algorithm. However, crl_longest was returned infrequently. For the Iris and Spambase datasets, Petal Length and crl_total had the greatest ranges in their respective datasets. However, Ca had the greatest range in the Glass dataset, and was returned a moderate amount. The most returned attribute, RI, had the smallest range. The histograms showing the frequency of each feature are shown in the appendix.

Discussion

Dataset	Total Features Returned	Most Returned Feature	Second Most Returned Feature
Iris	440	150 (34%)	116 (29%)
Glass	343	60 (17%)	55 (16%)
Spambase	404	13 (3%)	13 (3%)

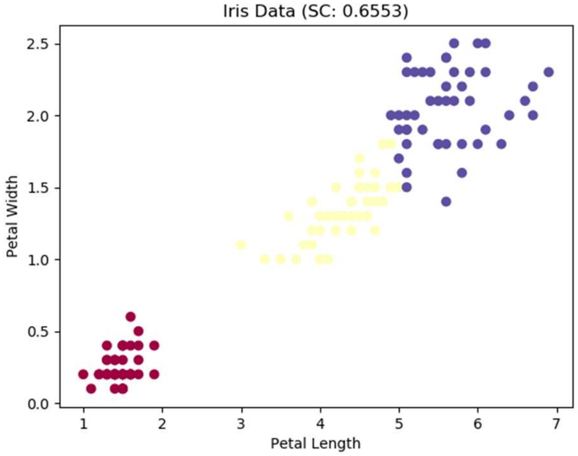
The K-Means algorithm consistently resulted in silhouette coefficients close to 1, and never produced a coefficient less than zero. Its performance did vary based on distribution, with the Iris coefficients taking the shape of a normal distribution with a mean around 0.7, the Spam coefficient trending closer to 1, and the Glass coefficients showing a wider distribution with peaks at 0.6 and 0.8.

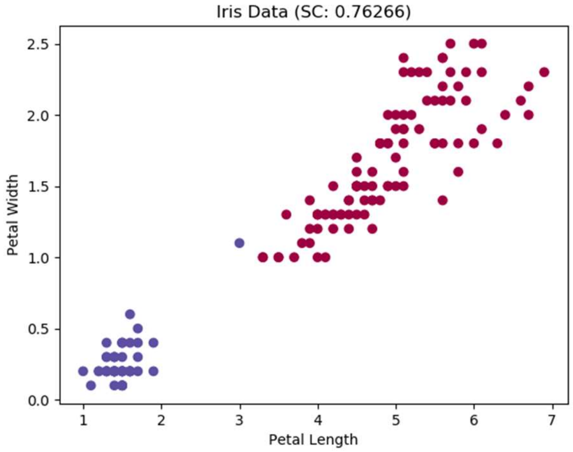
The SFS algorithm was less consistent, and while it favored certain attributes in the Iris dataset, in the Glass and Spambase datasets, there were a greater variety of features returned.

Summary

SFS is a greedy algorithm, and this was accounted for by randomly shuffling the order in which it operated on the features of a dataset. However, there were some factors that were not well controlled, including the fact that the less consistent datasets had more attributes to begin with. The frequencies of feature return should have been normalized across the probability of the feature being chosen at random, rather than simply counted. Additionally, features were counted when they were returned at all, with no extra classification regarding the other features they were returned with. Results where SFS returned a single feature should have likely been discarded, as it is unlikely that these datasets collapse to a single dimension so easily. As it stands, SFS was only successful in extracting the important features of the Iris

dataset, or the dataset with the fewest features. Greater average ranges of the attributes in the datasets was not associated with consistent feature extraction, indicating that while they may have been easier to cluster, the silhouette coefficient was not necessarily the best measure for those features.
Additionally, a high silhouette coefficient does not necessarily mean that proper clustering occurred.



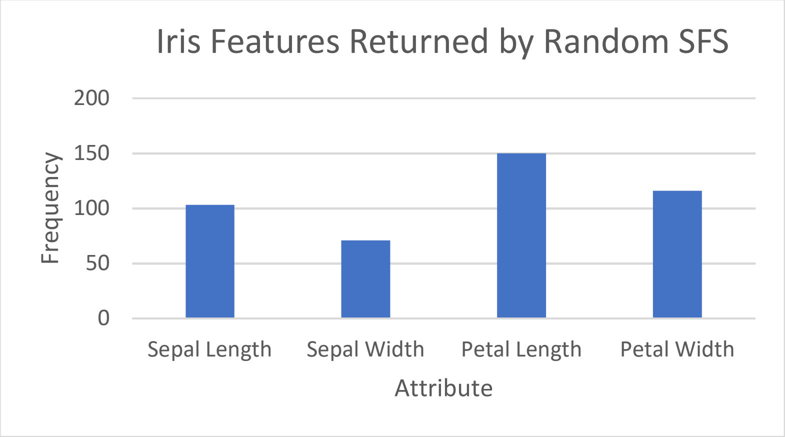


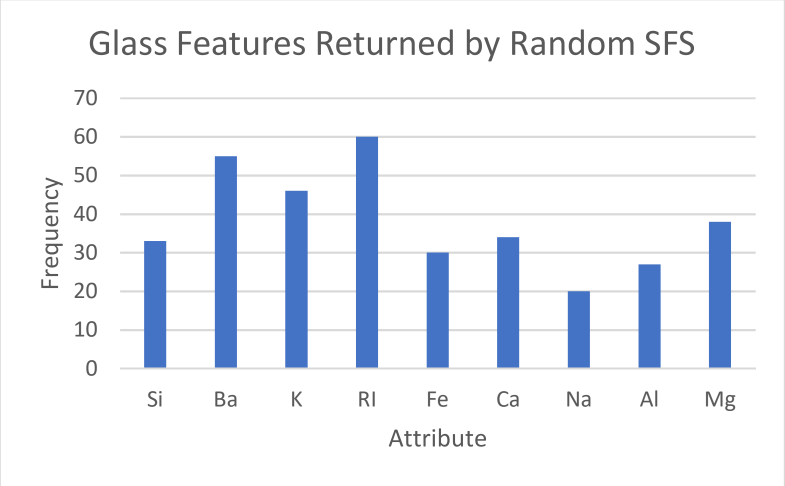
These two plots of the Iris dataset, plotted by the most extracted features, demonstrate that the silhouette coefficient is a measure of well-separated clusters, not correct ones. The plot with the correct number of clusters has a lower coefficient. Finally, Spambase is a large dataset, and the inconsistencies in the data are likely due to the relatively small sample sizes in each run. Given more optimized code and/or more time to run the program, the results would likely be different.
Overall, both algorithms performed well. K-means never produced an average silhouette coefficient less than 0.4, and most of the results clustered between 0.6 and 0.9. SFS extracted useful features in some cases, though it was not controlled well enough to do so in all cases.

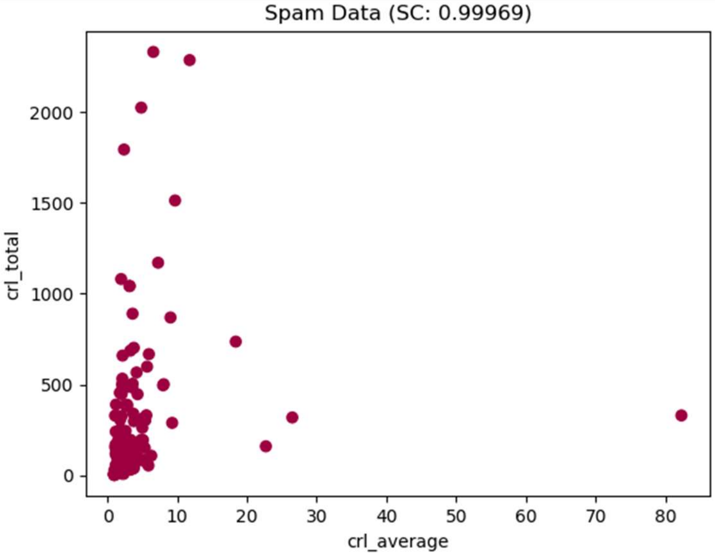
References

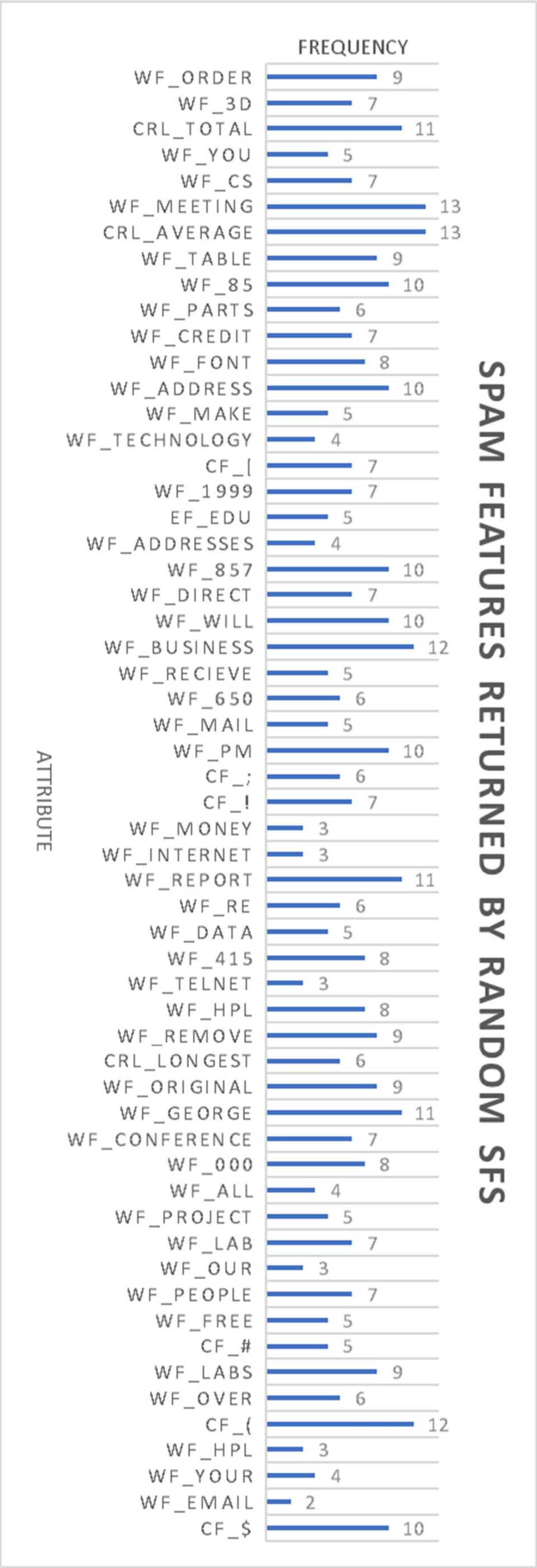
1) Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
2) Kohavi, R., and G. John. (1997). "Wrappers for Feature Subset Selection." Artificial Intelligence 97:273–324
3) MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press. pp. 281–297
4) Creator: R.A. Fisher Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov) Date: July, 1988
5) Creator: B. German -- Central Research Establishment Home Office Forensic Science Service Aldermaston, Reading, Berkshire RG7 4PN Date: September, 1987
6) Creators: Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304 Donor: George Forman (gforman at nospam hpl.hp.com) 650-857-7835
7) Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics. 20: 53–65

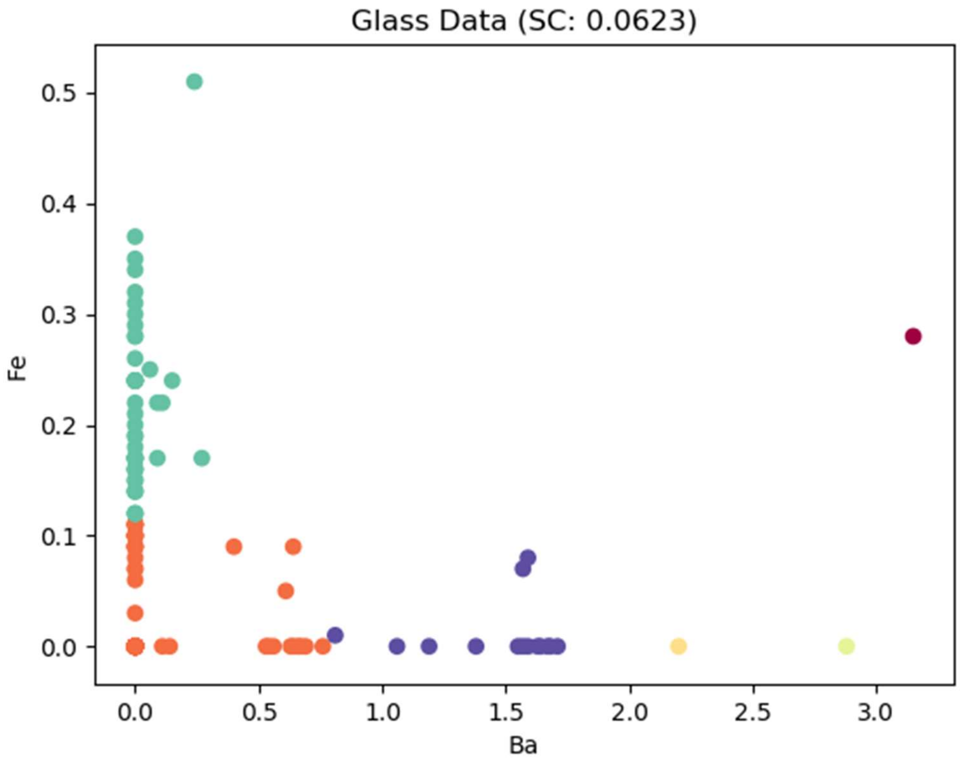
Appendix:

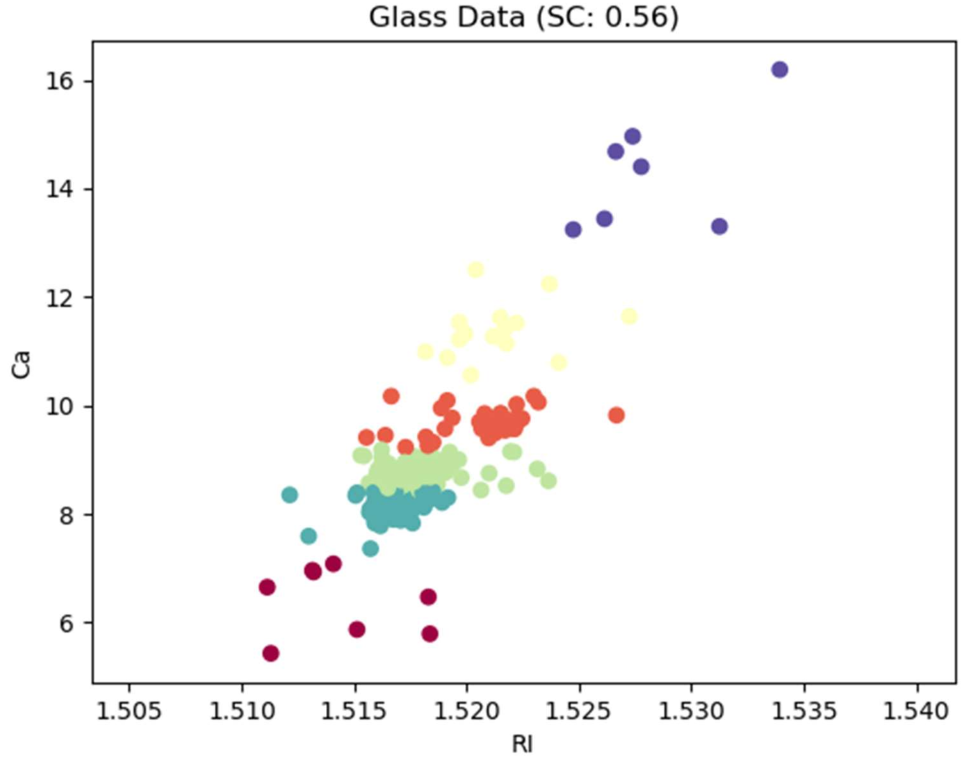

Iris Features Returned by Random SFS
Frequency
200
150
100
50
0
Sepal Length
Sepal Width
Petal Length
Petal Width
Attribute


Glass Features Returned by Random SFS
Frequency
70
60
50
40
30
20
10
0
Si
Ba
K
RI
Fe
Ca
Na
Al
Mg
Attribute


Spam Data (SC: 0.99969)
crl_total
2000
1500
1000
500
0
0
10
20
30
40
50
60
70
80
crl_average


SPAM FEATURES RETURNED BY RANDOM SFS
FREQUENCY
ATTRIBUTE
WF_ORDER 9
WF_3D 7
CRL_TOTAL 11
WF_YOU 5
WF_CS 7
WF_MEETING 13
CRL_AVERAGE 13
WF_TABLE 9
WF_85 10
WF_PARTS 6
WF_CREDIT 7
WF_FONT 8
WF_ADDRESS 10
WF_MAKE 5
WF_TECHNOLOGY 4
CF_[7
WF_1999 7
EF_EDU 5
WF_ADDRESSES 4
WF_857 10
WF_DIRECT 7
WF_WILL 10
WF_BUSINESS 12
WF_RECIEVE 5
WF_650 6
WF_MAIL 5
WF_PM 10
CF_; 6
CF_! 7
WF_MONEY 3
WF_INTERNET 3
WF_REPORT 11
WF_RE 6
WF_DATA 5
WF_415 8
WF_TELNET 3
WF_HPL 8
WF_REMOVE 9
CRL_LONGEST 6
WF_ORIGINAL 9
WF_GEORGE 11
WF_CONFERENCE 7
WF_000 8
WF_ALL 4
WF_PROJECT 5
WF_LAB 7
WF_OUR 3
WF_PEOPLE 7
WF_FREE 5
CF_# 5
WF_LABS 9
WF_OVER 6
CF_(12
WF_HPL 3
WF_YOUR 4
WF_EMAIL 2
CF_$ 10


Glass Data (SC: 0.0623)


Glass Data (SC: 0.56)