1. *Feature Selection.* Given the dataset "features.csv" (available in TeachCenter), which comprises 20 features and one target variable, select the most informative features for the target. Use a feature selection method of your choice.

   (a) Describe preprocessing and feature transformations steps if you made any (e.g., Did you create new features? Did you normalize the data?). Max. three sentences.

   (b) What is your feature selection method and why? Again max. three sentences.

   (c) Give ranking of your features together with the scores you got for them.

Answer (a) - Preprocessing & feature transformations:

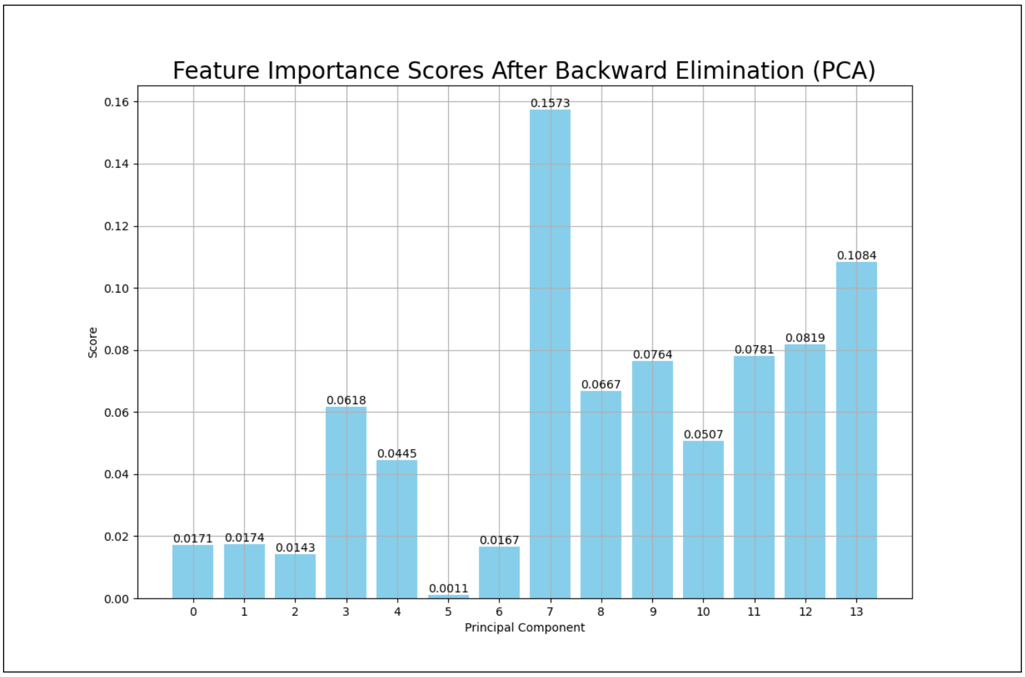
- Checked for null, nan and empty values. Found none.
- Outlier detection. visually first via scatter plot and then using KNN.
- Data was then standardtised before applying PCA with a 95% variance retention to further remove non-impactfull features. Was left with 14 features.
- Data was then divided using a typical 80/20 split (80% for training, 20% for testing)

Answer (b) - Feature selection method: Used 3 different methods.

- Backwards Elimination because - Removes less significative features interatively to simplify the model, thus simplifing it by only keeping the features that contribute most to the prediction.
- Recursive Feature Elimination (RFE) - Recursively removing one feature or a small set of them and making the model with the rest. It then ranks the features based on the model's performance.
- Univariate Selection - Tests features based on their relationship with the target variable using statistical tests. Straightforward approach that identifies the most impactfull features.

Answer (c) - Feature ranking:

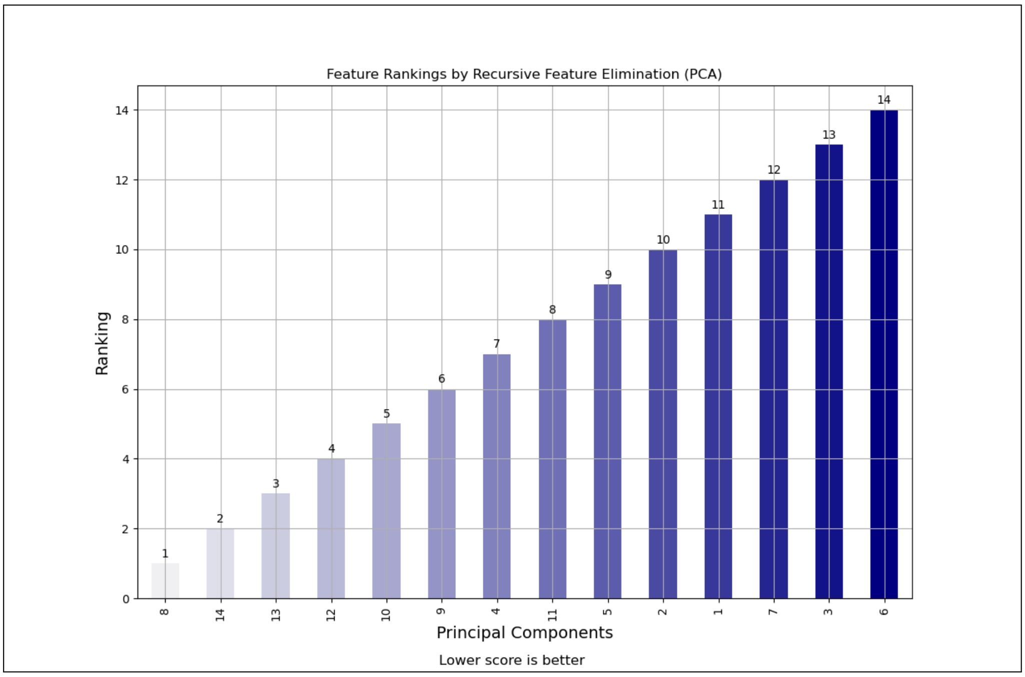
- Backwards Elimination:



Principal Component: 8, Score: 0.15732100986091457
Principal Component: 14, Score: 0.1083525472281407
Principal Component: 13, Score: 0.08187970795315384
Principal Component: 12, Score: 0.07807827153074007
Principal Component: 10, Score: 0.0763991832305842
Principal Component: 9, Score: 0.06670891885720762
Principal Component: 4, Score: 0.06178155968518729

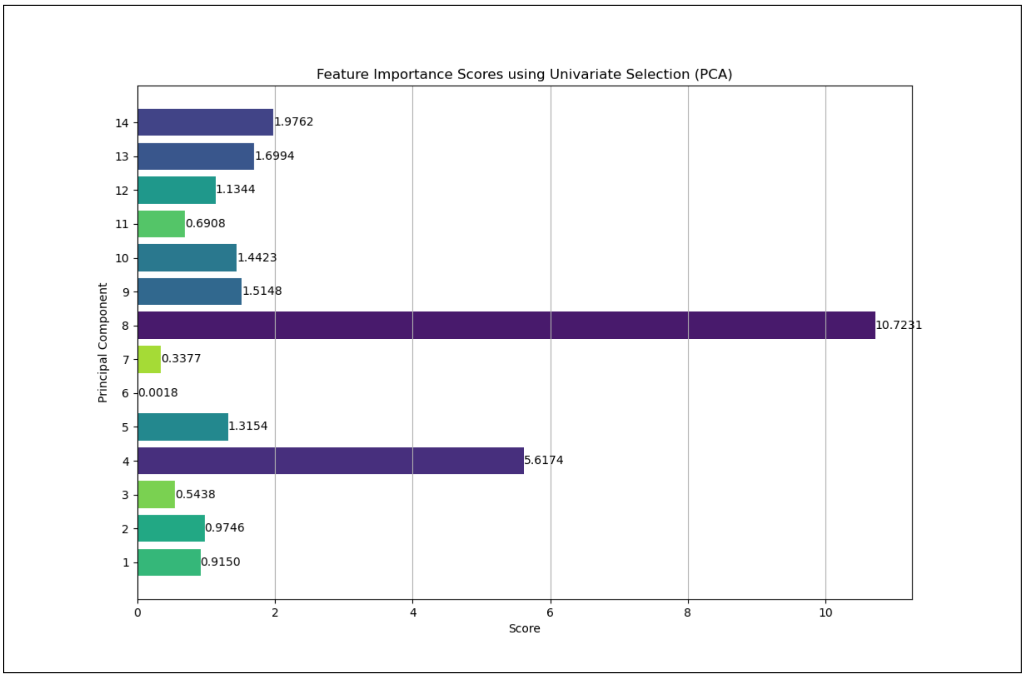
```
Principal Component: 11, Score: 0.05071224441181858
Principal Component: 5, Score: 0.04452588363972378
Principal Component: 2, Score: 0.01738185154209089
Principal Component: 1, Score: 0.01714112762317936
Principal Component: 7, Score: 0.016725782556327062
Principal Component: 3, Score: 0.014288574855603722
Principal Component: 6, Score: 0.001124926989765386
```

- RFE - Recursive Feature Elimination:



```
Principal Component: 8, Ranking: 1
Principal Component: 14, Ranking: 2
Principal Component: 13, Ranking: 3
Principal Component: 12, Ranking: 4
Principal Component: 10, Ranking: 5
Principal Component: 9, Ranking: 6
Principal Component: 4, Ranking: 7
Principal Component: 11, Ranking: 8
Principal Component: 5, Ranking: 9
Principal Component: 2, Ranking: 10
Principal Component: 1, Ranking: 11
Principal Component: 7, Ranking: 12
Principal Component: 3, Ranking: 13
Principal Component: 6, Ranking: 14
```

- Univariate:



Principal Component: 8, Score: 10.7231

Principal Component: 4, Score: 5.6174

Principal Component: 14, Score: 1.9762

Principal Component: 9, Score: 1.5148

Principal Component: 10, Score: 1.4423

Principal Component: 13, Score: 1.6994

Principal Component: 5, Score: 1.3154

Principal Component: 12, Score: 1.1344

Principal Component: 2, Score: 0.9746

Principal Component: 1, Score: 0.9150

Principal Component: 7, Score: 0.3377

Principal Component: 3, Score: 0.5438

Principal Component: 11, Score: 0.6908

Principal Component: 6, Score: 0.0018

2. *Dimensionality Reduction.* Given the previous dataset “features.csv” perform dimensionality reduction with PCA on the features (i.e., ignore the target variable in this task).

   (a) Describe preprocessing and feature transformations steps if you made any. Max. three sentences.

   (b) How many dimensions have you selected for this dataset. Explain how you selected them and why. Again max. three sentences.

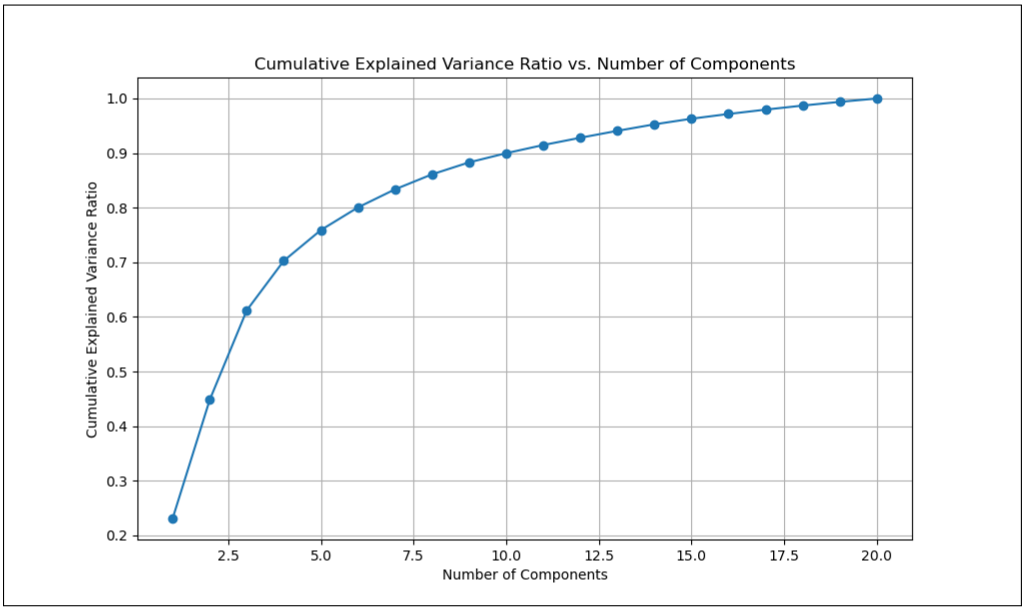
   (c) Give feature loadings for the dimensions you selected.

   (d) Interpret the dimensions that you selected. Again max. three sentences.

Answer (a) - Preprocessing & feature transformations:

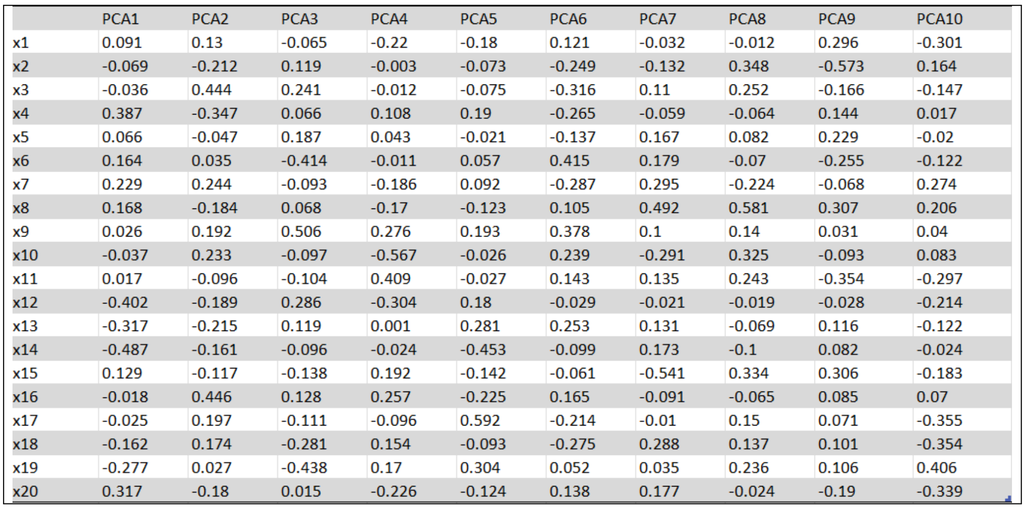
- Checked for null, nan and empty values. Found none.
- Outlier detection. visually first via scatter plot and then using KNN.
- Data was then standardtised to have a mean of 0 and standard deviation of 1.

Answer (b) - Selecting dimensions:



- The plot shows the cumulative variance of the depending on the number of components, meaning that for example, 10 components explain 0.9 of the data.
- Based on this plot, we should select a percentage that let’s us have a still high number of variance so our data has enough representation, anything after the ”elbow” of the curve is a good choice.
- Based on this previous considerations, i will select above the elbow point which is roughly at 0.8, so 0.9 should give us a good result, meaning 10 features were selected.

Answer (c) Feature Loadings:



	PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7	PCA8	PCA9	PCA10
x1	0.091	0.13	-0.065	-0.22	-0.18	0.121	-0.032	-0.012	0.296	-0.301
x2	-0.069	-0.212	0.119	-0.003	-0.073	-0.249	-0.132	0.348	-0.573	0.164
x3	-0.036	0.444	0.241	-0.012	-0.075	-0.316	0.11	0.252	-0.166	-0.147
x4	0.387	-0.347	0.066	0.108	0.19	-0.265	-0.059	-0.064	0.144	0.017
x5	0.066	-0.047	0.187	0.043	-0.021	-0.137	0.167	0.082	0.229	-0.02
x6	0.164	0.035	-0.414	-0.011	0.057	0.415	0.179	-0.07	-0.255	-0.122
x7	0.229	0.244	-0.093	-0.186	0.092	-0.287	0.295	-0.224	-0.068	0.274
x8	0.168	-0.184	0.068	-0.17	-0.123	0.105	0.492	0.581	0.307	0.206
x9	0.026	0.192	0.506	0.276	0.193	0.378	0.1	0.14	0.031	0.04
x10	-0.037	0.233	-0.097	-0.567	-0.026	0.239	-0.291	0.325	-0.093	0.083
x11	0.017	-0.096	-0.104	0.409	-0.027	0.143	0.135	0.243	-0.354	-0.297
x12	-0.402	-0.189	0.286	-0.304	0.18	-0.029	-0.021	-0.019	-0.028	-0.214
x13	-0.317	-0.215	0.119	0.001	0.281	0.253	0.131	-0.069	0.116	-0.122
x14	-0.487	-0.161	-0.096	-0.024	-0.453	-0.099	0.173	-0.1	0.082	-0.024
x15	0.129	-0.117	-0.138	0.192	-0.142	-0.061	-0.541	0.334	0.306	-0.183
x16	-0.018	0.446	0.128	0.257	-0.225	0.165	-0.091	-0.065	0.085	0.07
x17	-0.025	0.197	-0.111	-0.096	0.592	-0.214	-0.01	0.15	0.071	-0.355
x18	-0.162	0.174	-0.281	0.154	-0.093	-0.275	0.288	0.137	0.101	-0.354
x19	-0.277	0.027	-0.438	0.17	0.304	0.052	0.035	0.236	0.106	0.406
x20	0.317	-0.18	0.015	-0.226	-0.124	0.138	0.177	-0.024	-0.19	-0.339

Answer (d) We can generalise and say that whenever there are high value loadings with features it indicates that exists a strong correlation between original feature and the PCA component at test. When those values are negative it means that the relationship is inverse meaning that whenever the variables increase, the PCA component decreases.

- PCA-1 - x4 and x9 have high values and x13 and x14 have high negative values
- PCA-2 - x3 and x16 are the highest. contains relatively low number of negatives correlations
- PCA-3 - x3 is the highest correlation but it's comparatively low when looked at other PCA's
- PCA4 - similar to PCA3, the highest correlation is with x9 and it's comparatively low. contains a high negative correlation with the feature x10
- PCA5 - contains relatively high correlation with feature x17 of 0.592. Also very low negative values.
- PCA6 - Contains several moderately negative correlation (x2, x3, x4, x7, x8)
- PCA7 - Contains a highly positive correlation (x8) and a highly negative correlation (x15) almost symetrical in fact (0.492 vs -0.541)
- PCA8 - Contains a low number of negative correlations, and those that exist are low in value.
- PCA9 - x18 and x19 are very similar. Contains three moderately negatively correlated values (x2, x6, x11)
- PCA10 - Similarly to PCA9, contains the highest number of negatively correlated values, and those have also have moderately high values (x1, x11, x12, x13, x15, x17, x18, x20). Interesting to note that x11, x12, x13 and x15 have values that are quite close to eachother.

3. *Classification.* Given the dataset “class.csv” (available in TeachCenter), which comprises 20 features and one binary target variable, implement a classifier of your choice. Evaluate your classifier by a metric of your choice. If your model has hyperparameters cross-validate.

   (a) Describe preprocessing and feature transformations steps if you made any. Max. three sentences.

   (b) What is your model and why? Again max. three sentences.

   (c) Describe your evaluation setup. Max. one sentence.

   (d) Describe hyperparameter optimization if any. Give the final values of hyperparameters. Max. three sentence.

   (e) Give your evaluation results as text or a table.

Answer (a) - Preprocessing & feature transformations:

- Checked for null, nan and empty values. Found none.
- Outlier detection. visually first via scatter plot and then using KNN.
- Data was then standardtised before applying PCA with a 95% variance retention to further remove non-impactfull features.
- Data was divided into train/test data with a typical 80/20 split.

Answer (b) - Model choice:

- Random forest was chosen as the classifier algorithim because it is highly accurate and captures linear and non-linear relationships. It’s also less prone to overfitting, and since the dataset is not large, it’s not computationally expensive. A more simple decision tree algorithim was also briefly tested but had a lower accuracy of 0.845, while Random Forest has an accuracy of 0.895 and f1score = 0.9067.

Answer (c) - Evaluation setup:

- I went with Repeated k fold cross validation (k=5) since it works by repeatidly spliting the data into test/training data, training the model and evaluating it’s performance on it. (f1score was also done since it is simple and very popular)

Answer (d) - Hyperparameters:

- Using manual Grid searcht with a predetermined set of values, with the f1score as the scoring evaluation method. The hyperparametrs grid looked like:
  ’n-estimators’: [50, 100, 200],
  ’max-depth’: [None, 10, 20],
  ’max-features’: [’auto’, ’sqrt’, ’log2’]
  The best parameters were:
  Best Hyperparameters: ’max-depth’: 20, ’max-features’: ’sqrt’, ’n-estimators’: 200
- A Bayesian search was also performed with the parameters:
  param-space =
  ’n-estimators’: (50, 200),
  ’max-depth’: (1, 20)
  The best parameters were:
  Best Hyperparameters: OrderedDict([(’max-depth’, 13), (’n-estimators’, 200)])

Answer (e) - Results:

Metric	**Single Run**	**Mean (5-Fold CV)**
Accuracy	0.8950	0.9033
Precision	0.8947	0.9053
Recall	0.9189	0.9033
F1-score	0.9067	0.9032
ROC-AUC score	-	0.9034

Table 1: Random Forest Evaluation Metrics

Using 5-Fold Cross-Validation (Since $k = 5$, the data is split into multiple folds, thus the mean of each metric is a good representative number).

4. *Clustering.* Given the dataset "clustering.csv" (available in TeachCenter), which comprises 10 features, implement a clustering method of your choice.

   (a) Describe preprocessing and feature transformations steps if you made any. Max. three sentences.

   (b) What is your clustering algorithm and why? Again max. three sentences.

   (c) Use an internal evaluation metric to estimate the number of clusters. Plot the evaluation results as evaluation_metric vs. #clusters.

   (d) How many clusters did you select? Max. one sentence.

Answer (a) - Preprocessing & feature transformations:

- Checked for null, nan and empty values. Found none.
- Outlier detection. visually first via scatter plot and then using KNN.

Answer (b) - Clustering algorithm:

- I Decided to go with Hierarchical Clustering because of it's ability to capture complex shapes and structures including non-linear relationships.
- It also has a clear representation of the data as the output, leading to easy interpretation and understanding

Answer (c) - Evaluation plot: I will use Silhouete and the Calinski-Harabasz Index to have an idea on how many clusters should be selected.

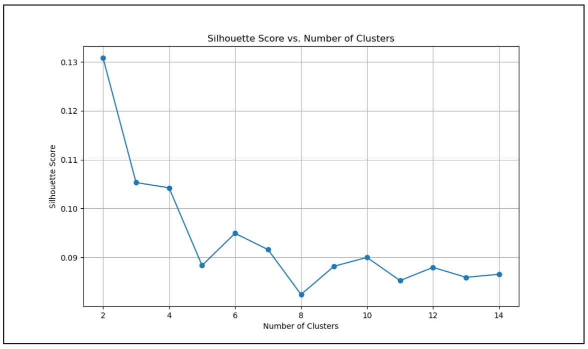


Figure 1: Silhouette Score

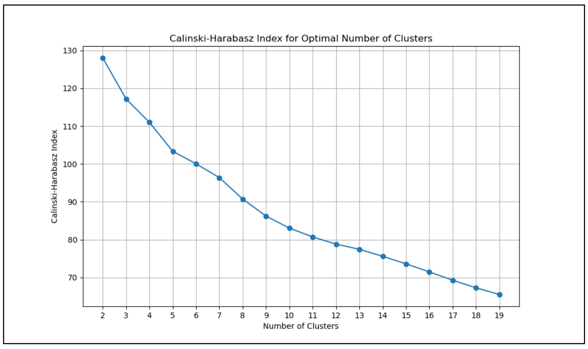
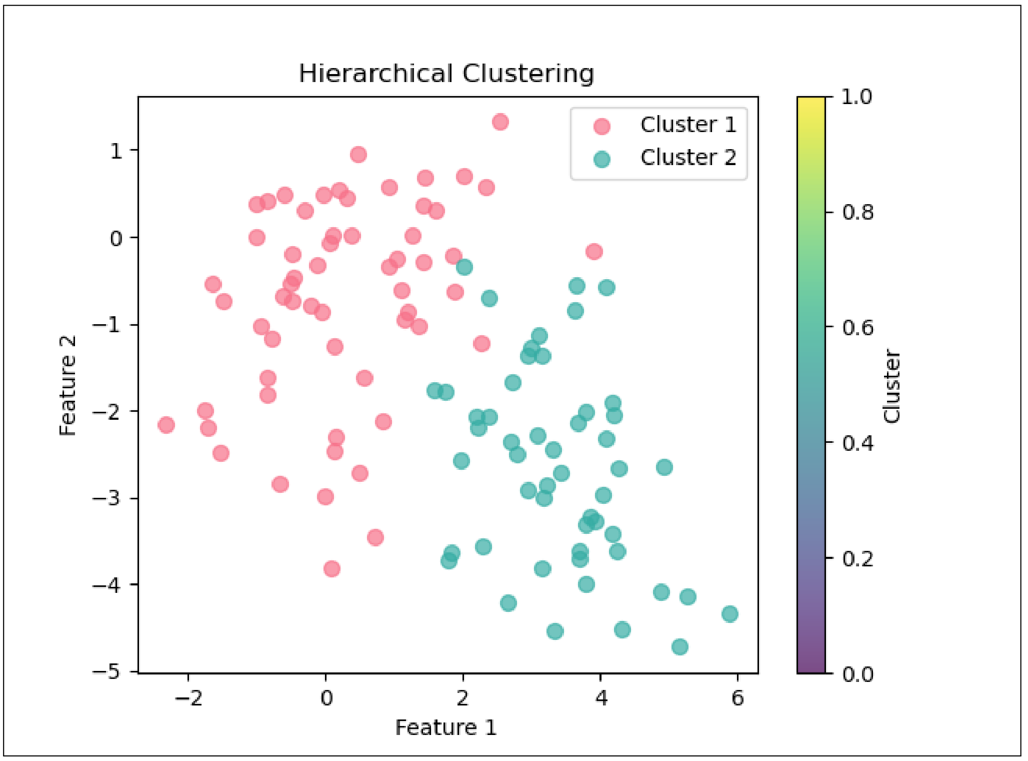


Figure 2: Calinski-Harabasz Index Score

Answer (d) - How many clusters:

- According to both Calinksi-Harabasz index and the Silhouete score, the best number of clusters seems to be 2, therefore 2 clusters were selected and plotted below.


Hierarchical Clustering
Cluster 1
Cluster 2
Feature 1
Feature 2
−2
0
2
4
6
1
0
−1
−2
−3
−4
−5
Cluster
1.0
0.8
0.6
0.4
0.2
0.0