Assignment 2 Report

By Maria Zhou

Contents

I. Suppervised - Classification (pancreaticCancer_dataset)

A. Project Code File

1. Project Code Link

https://colab.research.google.com/drive/119S3ch7tW3FFb1rI8zv760K_-ooRgp1q?usp=sharing

2. Project Goal

For this project, I will employ CART and CatBoost to predict pancreatic cancer diagnosis based on various features. Early detection of pancreatic cancer is crucial because it often presents no symptoms in its initial stages, and by the time symptoms appear, effective treatment options are very limited. Therefore, developing a predictive model for early diagnosis could significantly improve treatment outcomes and patient prognosis.

B. Dataset Information

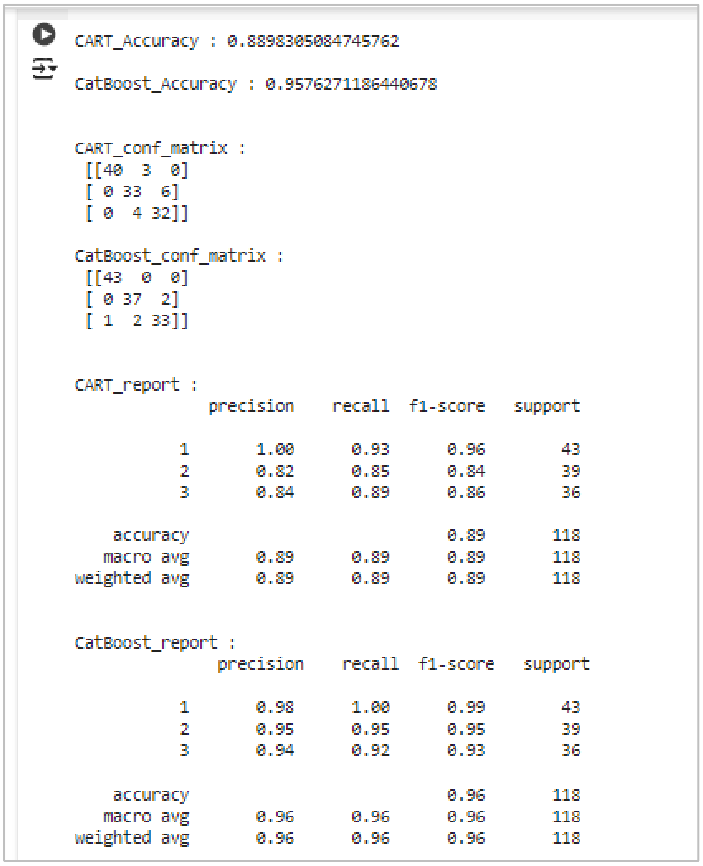
This dataset has 590 patients' records, including 183 Healthy controls patients (Class 1), 208 patients with non-cancerous pancreatic conditions(Class 2), and 199 patients with pancreatic cancer(Class 3).

Data information as follows:

```
RangeIndex: 590 entries, 0 to 589
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   sample_id                590 non-null    object
 1   patient_cohort           590 non-null    object
 2   sample_origin            590 non-null    object
 3   age                      590 non-null    int64
 4   sex                      590 non-null    object
 5   diagnosis                590 non-null    int64
 6   stage                    199 non-null    object
 7   benign_sample_diagnosis  208 non-null    object
 8   plasma_CA19_9            350 non-null    float64
 9   creatinine               590 non-null    float64
 10  LYVE1                    590 non-null    float64
 11  REG1B                    590 non-null    float64
 12  TFF1                     590 non-null    float64
 13  REG1A                    306 non-null    float64
```

C. Compare 2 Models (CART and CatBoost)

1. Evaluation Metrics



```
CART_Accuracy : 0.8898305084745762

CatBoost_Accuracy : 0.9576271186440678


CART_conf_matrix :
 [[40  3  0]
 [ 0 33  6]
 [ 0  4 32]]

CatBoost_conf_matrix :
 [[43  0  0]
 [ 0 37  2]
 [ 1  2 33]]


CART_report :
               precision    recall  f1-score   support

           1       1.00      0.93      0.96        43
           2       0.82      0.85      0.84        39
           3       0.84      0.89      0.86        36

    accuracy                           0.89       118
   macro avg       0.89      0.89      0.89       118
weighted avg       0.89      0.89      0.89       118


CatBoost_report :
                precision    recall  f1-score   support

           1       0.98      1.00      0.99        43
           2       0.95      0.95      0.95        39
           3       0.94      0.92      0.93        36

    accuracy                           0.96       118
   macro avg       0.96      0.96      0.96       118
weighted avg       0.96      0.96      0.96       118
```

Fig 1: Metrics Screenshot

a) *Accuracy:*

- CART Accuracy: 0.8898 (88.98%)
- CatBoost Accuracy: 0.9576 (95.76%)
- Result: CatBoost significantly outperforms CART in terms of accuracy, indicating its superior ability to correctly classify the instances in this dataset

b) *Confusion Matrices:*

- CART Confusion Matrix:
  - Class 1: 40 true positives, 3 false negatives, 0 false positives
  - Class 2: 33 true positives, 6 false negatives, 0 false positives
  - Class 3: 32 true positives, 4 false negatives, 0 false positives
- CatBoost Confusion Matrix:
  - Class 1: 43 true positives, 0 false negatives, 0 false positives
  - Class 2: 37 true positives, 2 false negatives, 0 false positives
  - Class 3: 33 true positives, 2 false negatives, 1 false positive
- Result: CatBoost shows better performance with no false positives for Class 1 and a very low number of false negatives across all classes, suggesting higher reliability in predictions.

2. Classification Reports

- CART Classification Report:
  - Class 1: Precision = 1.00, Recall = 0.93, F1-score = 0.96
  - Class 2: Precision = 0.82, Recall = 0.85, F1-score = 0.84
  - Class 3: Precision = 0.84, Recall = 0.89, F1-score = 0.86
- CatBoost Classification Report:
  - Class 1: Precision = 0.98, Recall = 1.00, F1-score = 0.99
  - Class 2: Precision = 0.95, Recall = 0.95, F1-score = 0.95
  - Class 3: Precision = 0.94, Recall = 0.92, F1-score = 0.93
- Result: CatBoost outperforms CART across all metrics. For Class 1, CatBoost achieves near-perfect precision and recall, resulting in a very high F1-score. Class 2 and Class 3 also show substantial improvements in precision, recall, and F1-score with CatBoost.

D. Conclusion

Based on these results, CatBoost outperforms CART in predicting pancreatic cancer diagnosis. CatBoost demonstrates higher accuracy, better precision, recall, and F1-scores across all classes. The confusion matrix of CatBoost shows fewer misclassifications, indicating more reliable and robust performance. Given these results, CatBoost is the preferred model for this dataset and application.

II. Suppervised - Regression(house_dataset)

A. Project Code File

1. Project Code Link

https://colab.research.google.com/drive/1etSnRVov-EI1PMfjv5a7MKFVAtXr_Bur?usp=sharing

2. Project Goal

For this project I will use SGDRegressor and LinearRegression to predict house prices based on its size, bedroom number, floors number, and house age. Then compare the predict prices with the real original prices to evaluate these 2 models’ performance.

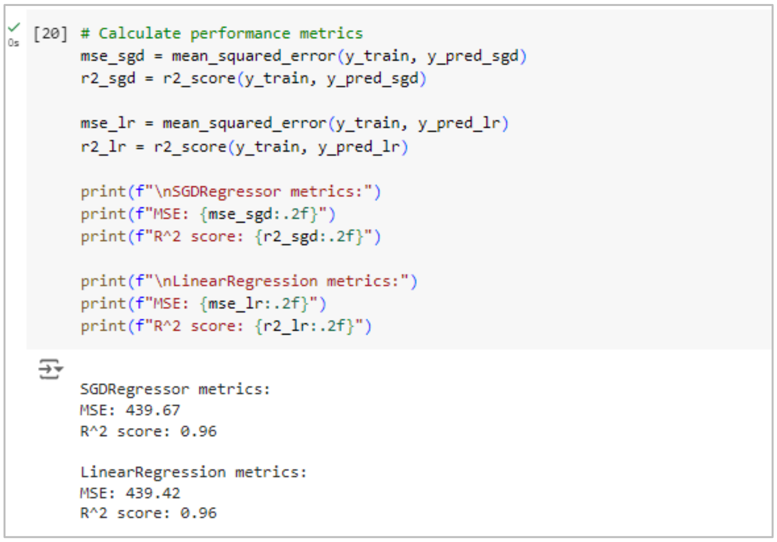
B. Dataset Information

Data information as follows:

```
RangeIndex: 100 entries, 0 to 99
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   size(sqft)  100 non-null    float64
 1   bedrooms    100 non-null    float64
 2   floors      100 non-null    float64
 3   age         100 non-null    float64
 4   price       100 non-null    float64
```

C. Compare 2 Models (SGDRegressor and LinearRegression)

1. Evaluation Metrics



```
[20] # Calculate performance metrics
     mse_sgd = mean_squared_error(y_train, y_pred_sgd)
     r2_sgd = r2_score(y_train, y_pred_sgd)

     mse_lr = mean_squared_error(y_train, y_pred_lr)
     r2_lr = r2_score(y_train, y_pred_lr)

     print(f"\nSGDRegressor metrics:")
     print(f"MSE: {mse_sgd:.2f}")
     print(f"R^2 score: {r2_sgd:.2f}")

     print(f"\nLinearRegression metrics:")
     print(f"MSE: {mse_lr:.2f}")
     print(f"R^2 score: {r2_lr:.2f}")
```

```

SGDRegressor metrics:
MSE: 439.67
R^2 score: 0.96

LinearRegression metrics:
MSE: 439.42
R^2 score: 0.96
```

Fig 1: Metrics Screenshot

a) *Mean Squared Error (MSE):*

- SGDRegressor: 439.67
- LinearRegression: 439.42
- Result: Both models have very similar MSE values, indicating that the average squared difference between the actual and predicted values is almost the same for both models. However, LinearRegression has a slightly lower MSE.

b) *R^2 Score:*

- SGDRegressor: 0.96
- LinearRegression: 0.96
- Result: Both models have the same R^2 score, indicating that they both explain 96% of the variance in the target variable, demonstrating excellent predictive performance.

D. Conclusion

Given the nearly identical performance of both models in terms of MSE and R^2 score, either model can be considered effective for predicting price based on this dataset. However, LinearRegression has a slight edge due to its marginally lower MSE.

If the slight difference in MSE is not critical to the application, it is OK to choose either model based on other considerations such as computational efficiency or ease of implementation. For instance, LinearRegression is simpler and easier to interpret, while SGDRegressor might be more scalable for very large datasets.

III. Unsupervised - Clustering(mallCustomers_dataset)

A. Project Code File

1. Project Code Link

https://colab.research.google.com/drive/1xI7qbq_DsAOzXvA_1tupJO06YHrLYMbC?usp=sharing

2. Project Goal

For this project, I will use Hierarchical Clustering and KMeans Clustering techniques to analyze and segment customers based on their annual income and spending score. By identifying distinct clusters, the mall can better understand customer behavior and tailor marketing strategies to target specific groups effectively. This approach will help in optimizing customer engagement and increasing sales.

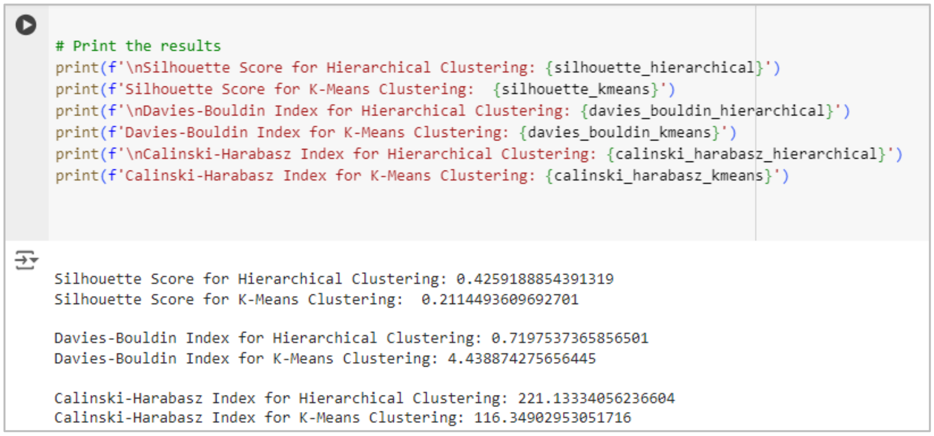
B. Dataset Information

Data information as follows:

```
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
```

C. Compare 2 Models (Hierarchical Clustering and KMeans Clustering)

1. Evaluation Metrics



```
# Print the results
print(f'\nSilhouette Score for Hierarchical Clustering: {silhouette_hierarchical}')
print(f'Silhouette Score for K-Means Clustering:  {silhouette_kmeans}')
print(f'\nDavies-Bouldin Index for Hierarchical Clustering: {davies_bouldin_hierarchical}')
print(f'Davies-Bouldin Index for K-Means Clustering: {davies_bouldin_kmeans}')
print(f'\nCalinski-Harabasz Index for Hierarchical Clustering: {calinski_harabasz_hierarchical}')
print(f'Calinski-Harabasz Index for K-Means Clustering: {calinski_harabasz_kmeans}')
```

```
Silhouette Score for Hierarchical Clustering: 0.4259188854391319
Silhouette Score for K-Means Clustering:  0.2114493609692701

Davies-Bouldin Index for Hierarchical Clustering: 0.7197537365856501
Davies-Bouldin Index for K-Means Clustering: 4.438874275656445

Calinski-Harabasz Index for Hierarchical Clustering: 221.13334056236604
Calinski-Harabasz Index for K-Means Clustering: 116.34902953051716
```

Fig 1: Metrics Screenshot

a) *Silhouette Score:*

- Hierarchical Clustering: 0.4259
- KMeans Clustering: 0.2114
- Result: The Silhouette Score indicates how similar an object is to its own cluster compared to other clusters. Higher scores indicate better-defined clusters. Hierarchical Clustering performs better with a score of 0.4259, compared to KMeans Clustering with a score of 0.2114.

b) *Davies-Bouldin Score:*

- Hierarchical Clustering: 0.7198
- KMeans Clustering: 4.4389
- Result: The Davies-Bouldin Index measures the average similarity ratio of each cluster with the cluster most like it. Lower values indicate better clustering quality. Hierarchical Clustering has a significantly lower Davies-Bouldin Index (0.7198) compared to KMeans Clustering (4.4389), suggesting that the clusters formed by Hierarchical Clustering are more distinct and compact.

c) *Calinski-Harabasz Score:*

- Hierarchical Clustering: 221.1333

- KMeans Clustering: 116.3490
- Result: The Calinski-Harabasz Index, also known as the Variance Ratio Criterion, evaluates the ratio of the sum of between-cluster dispersion to within-cluster dispersion. Higher values indicate better-defined clusters. Hierarchical Clustering again outperforms KMeans Clustering with a score of 221.1333 versus 116.3490.

D. Conclusion

Based on the evaluation metrics, Hierarchical Clustering demonstrates superior performance compared to KMeans Clustering in this customer segmentation task. Hierarchical Clustering produces better-defined and more distinct clusters, as indicated by higher Silhouette Scores, lower Davies-Bouldin Indices, and higher Calinski-Harabasz Indices. It is recommended to use Hierarchical Clustering for segmenting customers based on their annual income and spending score. The insights gained from this clustering can help the mall develop targeted marketing strategies, enhancing customer engagement and optimizing sales efforts.

IV. Unsupervised - Clustering(colleges_dataset)

A. Project Code File

1. Project Code Link

https://colab.research.google.com/drive/1HbVpyfFpk_4ghKt4Txs4wEOISuq_eYjK?usp=sharing

2. Project Goal

For this project, I will use Hierarchical Clustering and KMeans Clustering to divide college dataset into 2 clusters (privacy school and public school) because of knowing the ‘Private’ feature (‘Yes’ – privacy, ‘No’ – public, in most of time, this target is not known, here is for experiment). I will compare these 2 models’ performance by comparing matrics (silhouette_score, davies_bouldin_score, calinski_harabasz_score), and comparing their accuracy based on the real original ‘Private’ feature value.

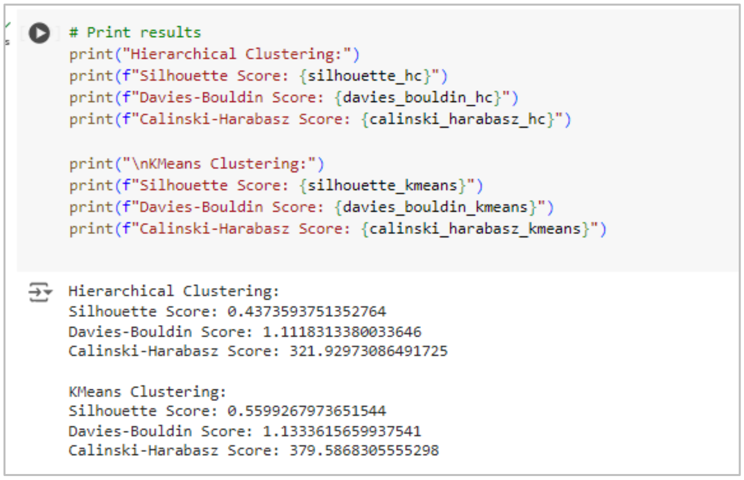
B. Dataset Information

Data information as follows:

```
Index: 777 entries, Abilene Christian University to York College of
Pennsylvania
Data columns (total 18 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Private      777 non-null    object
 1   Apps         777 non-null    int64
 2   Accept       777 non-null    int64
 3   Enroll       777 non-null    int64
 4   Top10perc    777 non-null    int64
 5   Top25perc    777 non-null    int64
 6   F.Undergrad  777 non-null    int64
 7   P.Undergrad  777 non-null    int64
 8   Outstate     777 non-null    int64
 9   Room.Board   777 non-null    int64
 10  Books        777 non-null    int64
 11  Personal     777 non-null    int64
 12  PhD          777 non-null    int64
 13  Terminal     777 non-null    int64
 14  S.F.Ratio    777 non-null    float64
 15  perc.alumni  777 non-null    int64
 16  Expend       777 non-null    int64
 17  Grad.Rate    777 non-null    int64
```

C. Compare 2 Models (Hierarchical Clustering and KMeans Clustering)

1. Evaluation Metrics



```python
# Print results
print("Hierarchical Clustering:")
print(f"Silhouette Score: {silhouette_hc}")
print(f"Davies-Bouldin Score: {davies_bouldin_hc}")
print(f"Calinski-Harabasz Score: {calinski_harabasz_hc}")

print("\nKMeans Clustering:")
print(f"Silhouette Score: {silhouette_kmeans}")
print(f"Davies-Bouldin Score: {davies_bouldin_kmeans}")
print(f"Calinski-Harabasz Score: {calinski_harabasz_kmeans}")
```

```
Hierarchical Clustering:
Silhouette Score: 0.4373593751352764
Davies-Bouldin Score: 1.1118313380033646
Calinski-Harabasz Score: 321.92973086491725

KMeans Clustering:
Silhouette Score: 0.5599267973651544
Davies-Bouldin Score: 1.1333615659937541
Calinski-Harabasz Score: 379.5868305555298
```

Fig 1: Metrics Screenshot

a) *Silhouette Score:*

- Hierarchical Clustering: 0.437
- KMeans Clustering: 0.560
- Result: KMeans shows a higher Silhouette Score, indicating better cluster cohesion and separation.

b) *Davies-Bouldin Score:*

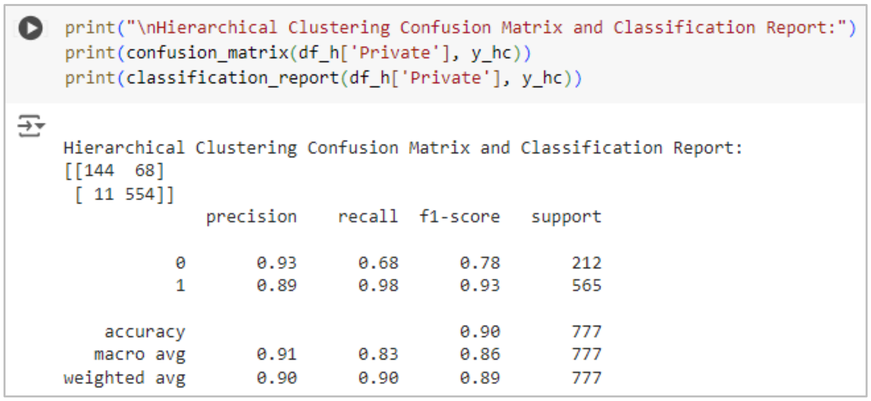
- Hierarchical Clustering: 1.112
- KMeans Clustering: 1.133
- Result: Hierarchical Clustering has a slightly better Davies-Bouldin Score, indicating potentially better cluster compactness and separation.

c) *Calinski-Harabasz Score:*

- Hierarchical Clustering: 321.930
- KMeans Clustering: 379.587
- Result: KMeans shows a higher Calinski-Harabasz Score, indicating better cluster density and separation.

2. Classification Reports

a) *Hierarchical Clustering*



```
print("\nHierarchical Clustering Confusion Matrix and Classification Report:")
print(confusion_matrix(df_h['Private'], y_hc))
print(classification_report(df_h['Private'], y_hc))
```

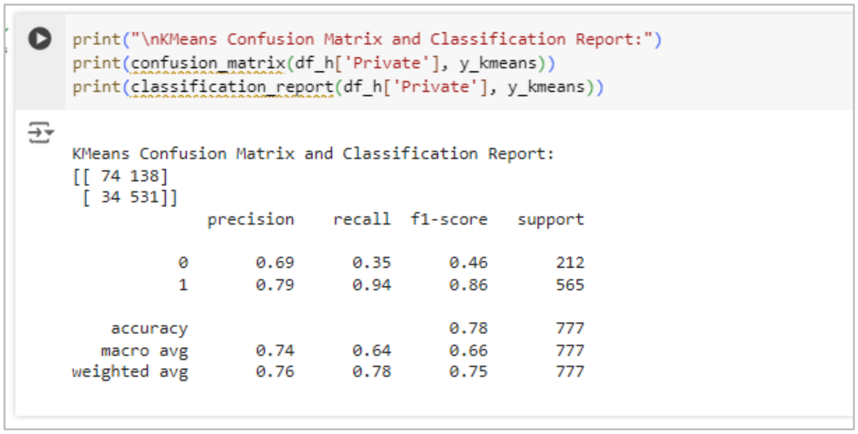
```
Hierarchical Clustering Confusion Matrix and Classification Report:
[[144  68]
 [ 11 554]]
              precision    recall  f1-score   support

           0       0.93      0.68      0.78       212
           1       0.89      0.98      0.93       565

    accuracy                           0.90       777
   macro avg       0.91      0.83      0.86       777
weighted avg       0.90      0.90      0.89       777
```

- Accuracy: 90%
- Precision (class 0/public colleges): 0.93
- Precision (class 1/private colleges): 0.89
- Recall (class 0/public colleges): 0.68
- Recall (class 1/private colleges): 0.98

b) *KMeans Clustering*



```
print("\nKMeans Confusion Matrix and Classification Report:")
print(confusion_matrix(df_h['Private'], y_kmeans))
print(classification_report(df_h['Private'], y_kmeans))
```

```
KMeans Confusion Matrix and Classification Report:
[[ 74 138]
 [ 34 531]]
              precision    recall  f1-score   support

           0       0.69      0.35      0.46       212
           1       0.79      0.94      0.86       565

    accuracy                           0.78       777
   macro avg       0.74      0.64      0.66       777
weighted avg       0.76      0.78      0.75       777
```

- Accuracy: 78%
- Precision (class 0/public colleges): 0.69
- Precision (class 1/private colleges): 0.79
- Recall (class 0/public colleges): 0.35
- Recall (class 1/private colleges): 0.94

c) *Summary*

- Accuracy: Hierarchical Clustering achieves higher accuracy (90%) compared to KMeans (78%).
- Precision and Recall: KMeans shows higher precision and recall for private colleges (class 1) but lower performance for public colleges (class 0). Hierarchical Clustering shows balanced precision and recall for both classes, with notably high precision for public colleges.

D. Conclusion

Based on these results, Hierarchical Clustering appears to be more suitable for the college dataset. Hierarchical Clustering generally performs better in terms of accuracy and balanced precision-recall scores across classes. However, KMeans shows better performance in terms of Silhouette Score and Calinski-Harabasz Score, indicating potentially better-defined clusters in terms of cohesion and separation. The choice between these two algorithms would depend on the specific goals of the analysis and the type of dataset.