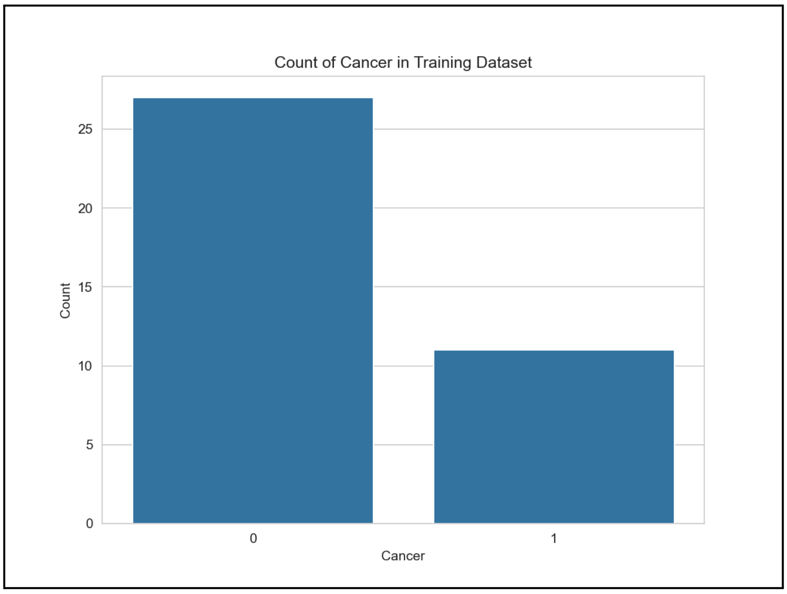
Assignment: Expression of Genes in AML and ALL type Leukemia

Q1) Build a classification model to predict the type of Leukemia as AML or ALL. Your classifier should be intepretable, I.e., one can look at the model and figure out how it identifies the classes, e.g., rule-based classifiers, logistic regression, or decision trees. You may use more complex models but then you will have to apply explainable AI methods as a separate step to interpret results. Tune model parameters to achieve the best possible prediction results but do not use the test set for this tuning.

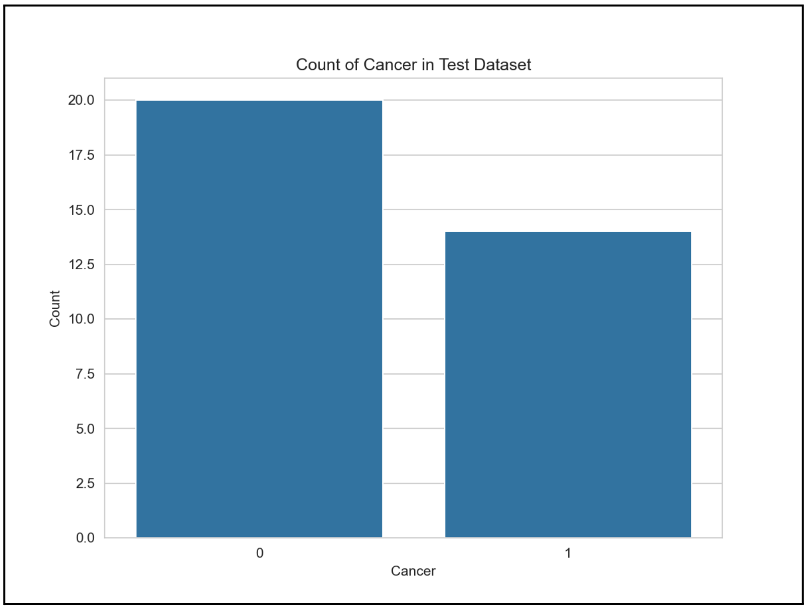
The dataset has already been divided into training and testing sets. Please use them for your training and testing purposes, respectively. Also, note that the data has already been normalized.

After pre-processing:

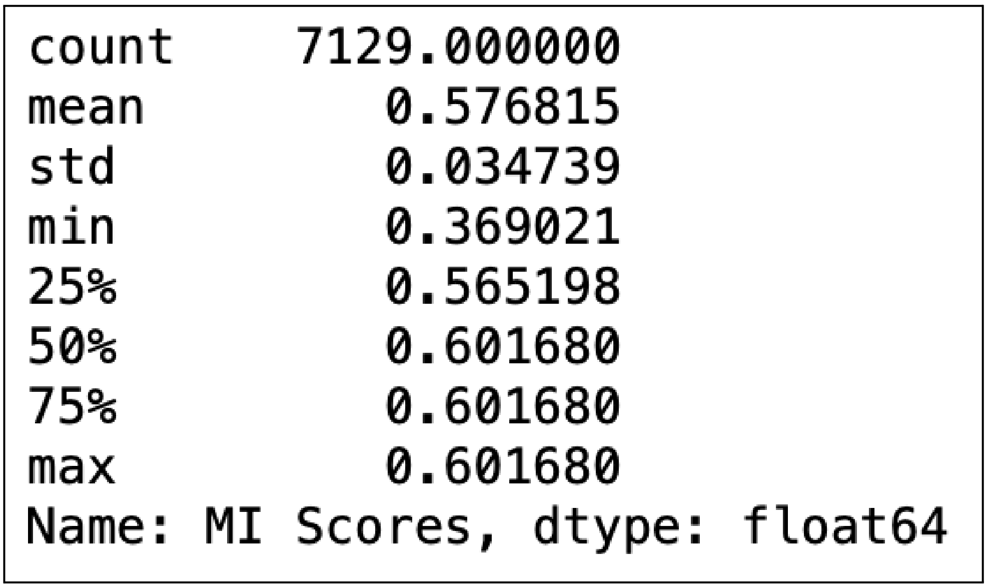
1. Training dataset: 38 samples with 7129 features



2. Test dataset: 34 samples with 7129 features



Distribution of the mutual information scores in the training dataset:



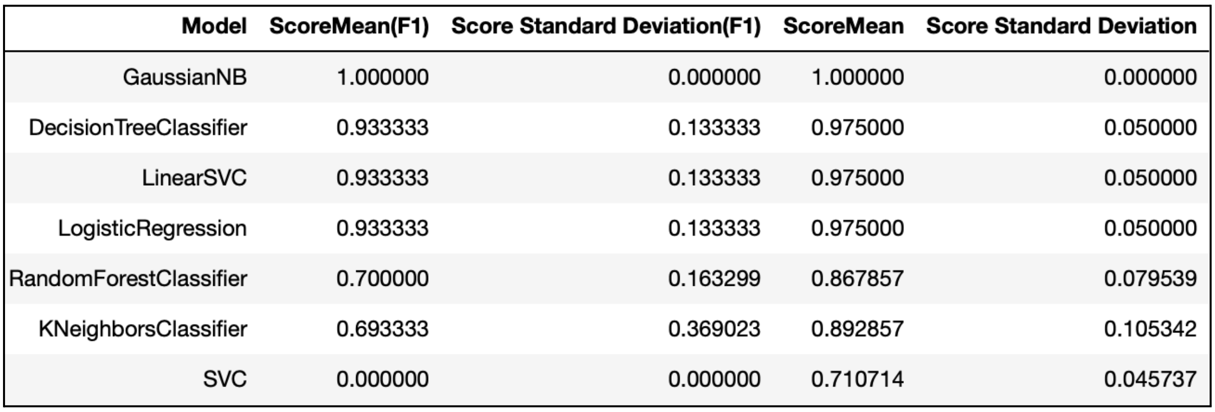
```
count    7129.000000
mean        0.576815
std         0.034739
min         0.369021
25%         0.565198
50%         0.601680
75%         0.601680
max         0.601680
Name: MI Scores, dtype: float64
```

Since the 1st quartile of the mutual information scores for all features in the training dataset is above 0.5, with the maximum around 0.6, it suggests that almost all features are equally important.

Following models were tested on the training dataset using stratified 5-fold cross-validation:

```
rs = 42, mi = 1000

models = [
    DecisionTreeClassifier(random_state=rs),
    KNeighborsClassifier(),
    GaussianNB(),
    LinearSVC(random_state=rs, max_iter=mi),
    SVC(random_state=rs, max_iter=mi),
    LogisticRegression(random_state=rs, max_iter=mi),
    RandomForestClassifier(random_state=rs),
]
```

Mean and standard deviation of F1 score and accuracy score after running cross-validation:



Model	ScoreMean(F1)	Score Standard Deviation(F1)	ScoreMean	Score Standard Deviation
GaussianNB	1.000000	0.000000	1.000000	0.000000
DecisionTreeClassifier	0.933333	0.133333	0.975000	0.050000
LinearSVC	0.933333	0.133333	0.975000	0.050000
LogisticRegression	0.933333	0.133333	0.975000	0.050000
RandomForestClassifier	0.700000	0.163299	0.867857	0.079539
KNeighborsClassifier	0.693333	0.369023	0.892857	0.105342
SVC	0.000000	0.000000	0.710714	0.045737

GaussianNB, DecisionTreeClassifier, LinearSVC, and LogisticRegression performed well considering both F1-score and accuracy score. However, DecisionTreeClassifier was chosen for its simpler rule-based interpretability.

Hyperparameter tuning: It was performed on the DecisionTreeClassifier using RandomizedSearchCV.

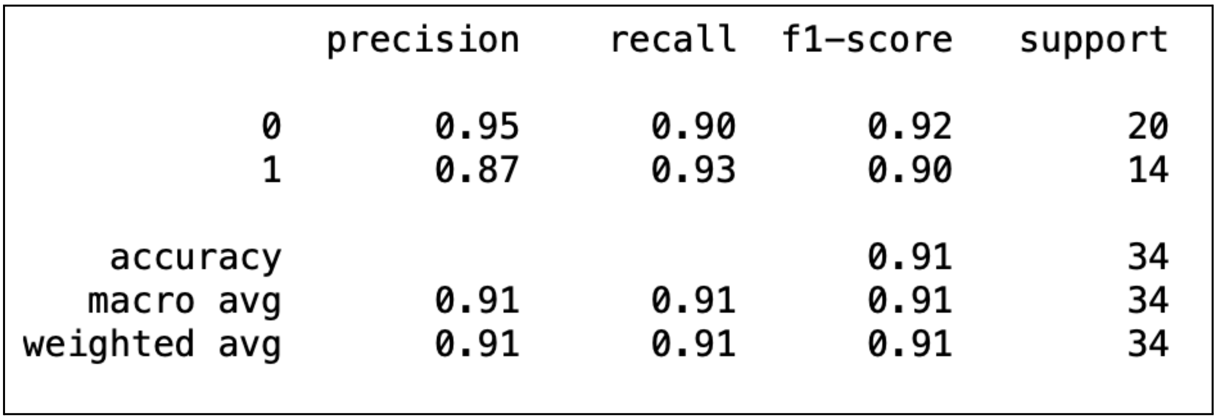
```
dtc_params = {
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random'],
    'max_depth': [None] + list(range(1, 51)),
    'min_samples_split': [2, 5, 10, 20],
    'min_samples_leaf': [1, 2, 4, 6, 8],
    'max_features': [None, 'auto', 'sqrt', 'log2'],
    'max_leaf_nodes': [None] + list(range(10, 101, 10)),
    'min_impurity_decrease': [0.0, 0.01, 0.02, 0.05, 0.1]
}

dtc_scv = RandomizedSearchCV(
    DecisionTreeClassifier(random_state=rs),
    param_distributions=dtc_params,
    n_iter=300,
    cv=5,
    verbose=2,
    random_state=rs,
    n_jobs=-1,
    scoring="f1",
)

dtc_scv.fit(train_X, train_y)
```

I found that the default parameters work fine in this scenario.

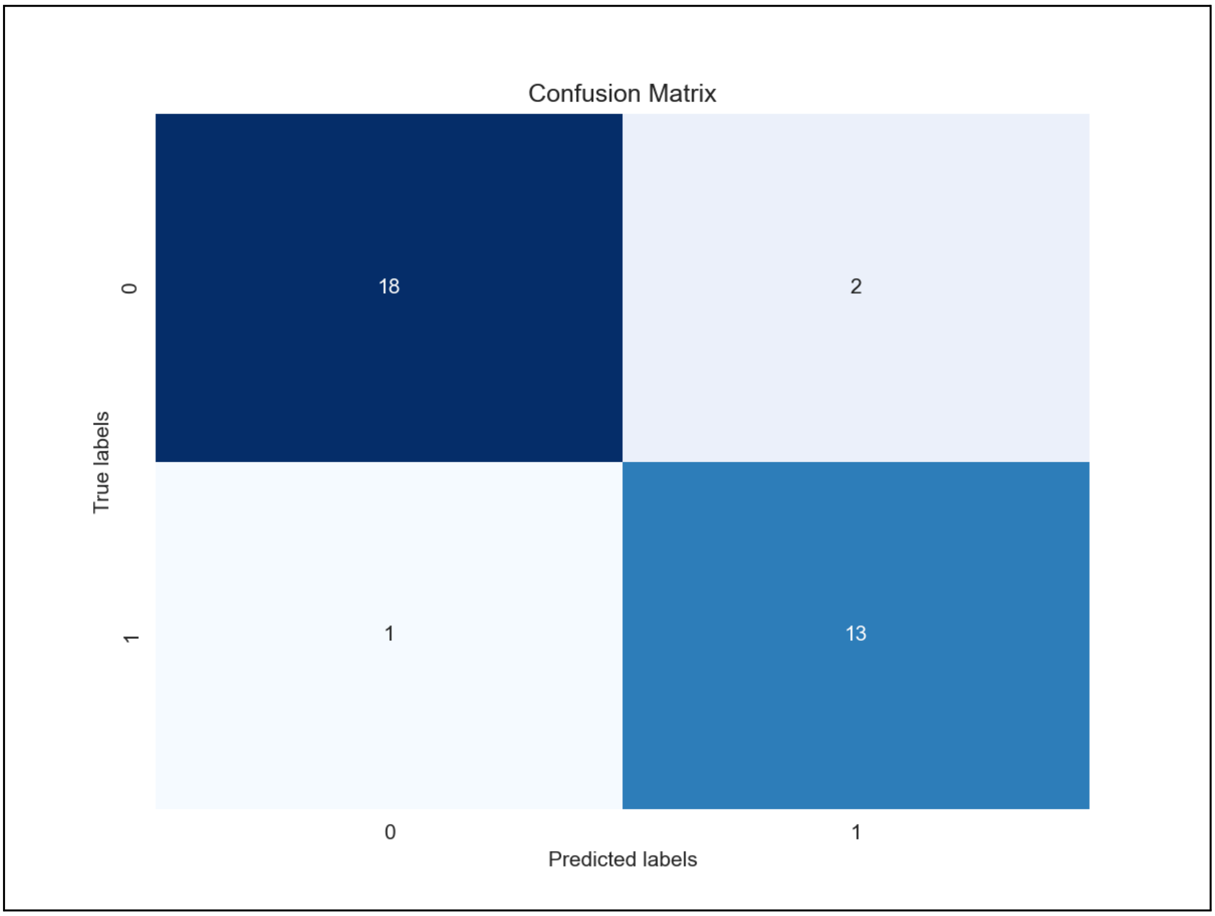
Model building:

1. Model: DecisionTreeClassifier with default parameters
2. Training F1-score and accuracy score: 1.0 and 1.0 respectively
3. Testing F1-score and accuracy score: 0.897 and 0.912 respectively

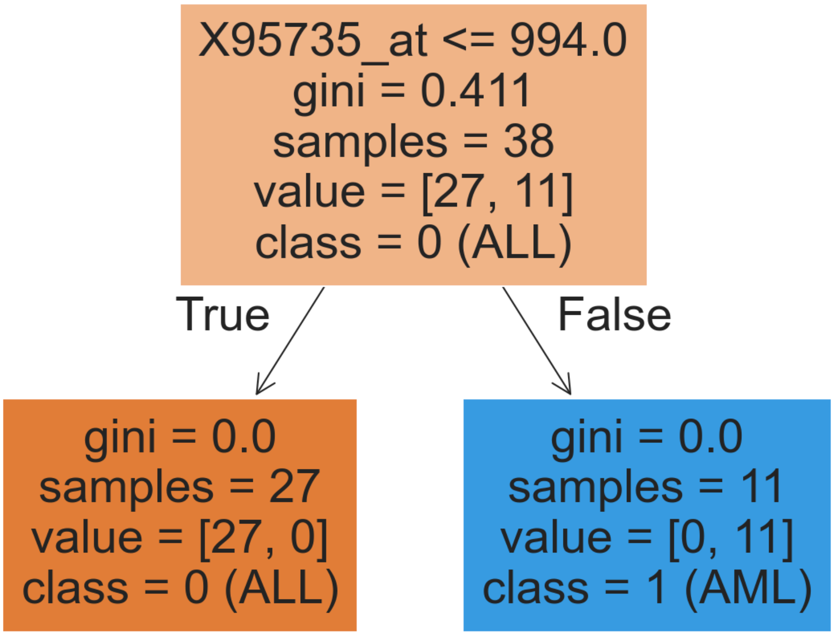
4. On test dataset:



```
              precision    recall  f1-score   support

           0       0.95      0.90      0.92        20
           1       0.87      0.93      0.90        14

    accuracy                           0.91        34
   macro avg       0.91      0.91      0.91        34
weighted avg       0.91      0.91      0.91        34
```



Q2) Using the model you developed in Q1, identify any patterns in gene expression that help distinguish between the types AML and ALL.

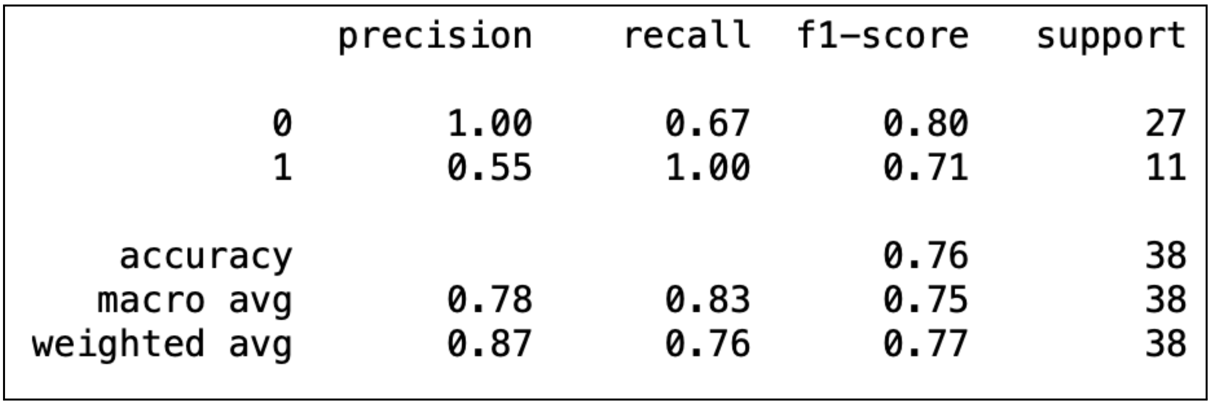


Above ruled-based pattern had been noticed on:

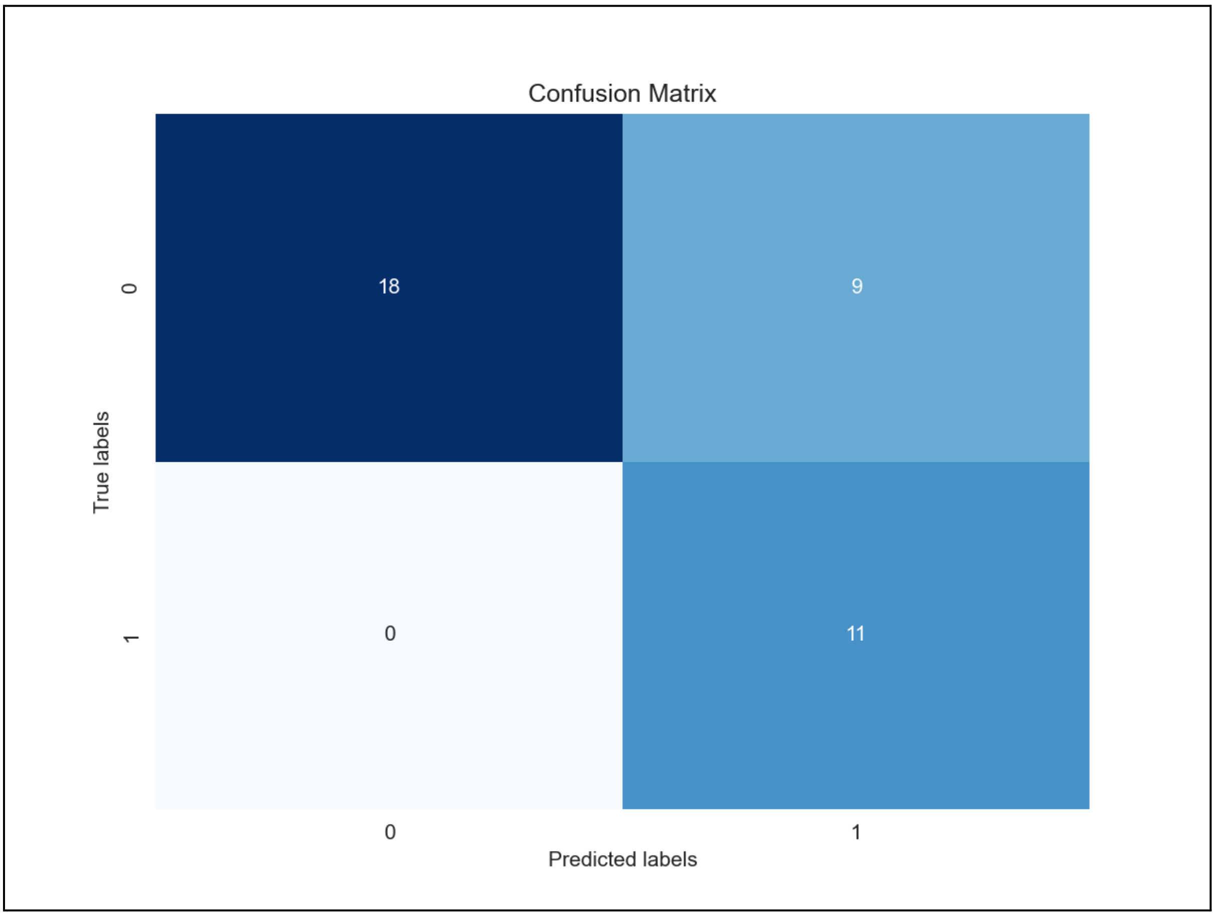
- Gene Description: Zyxin
- Gene Accession Number: X95735_at

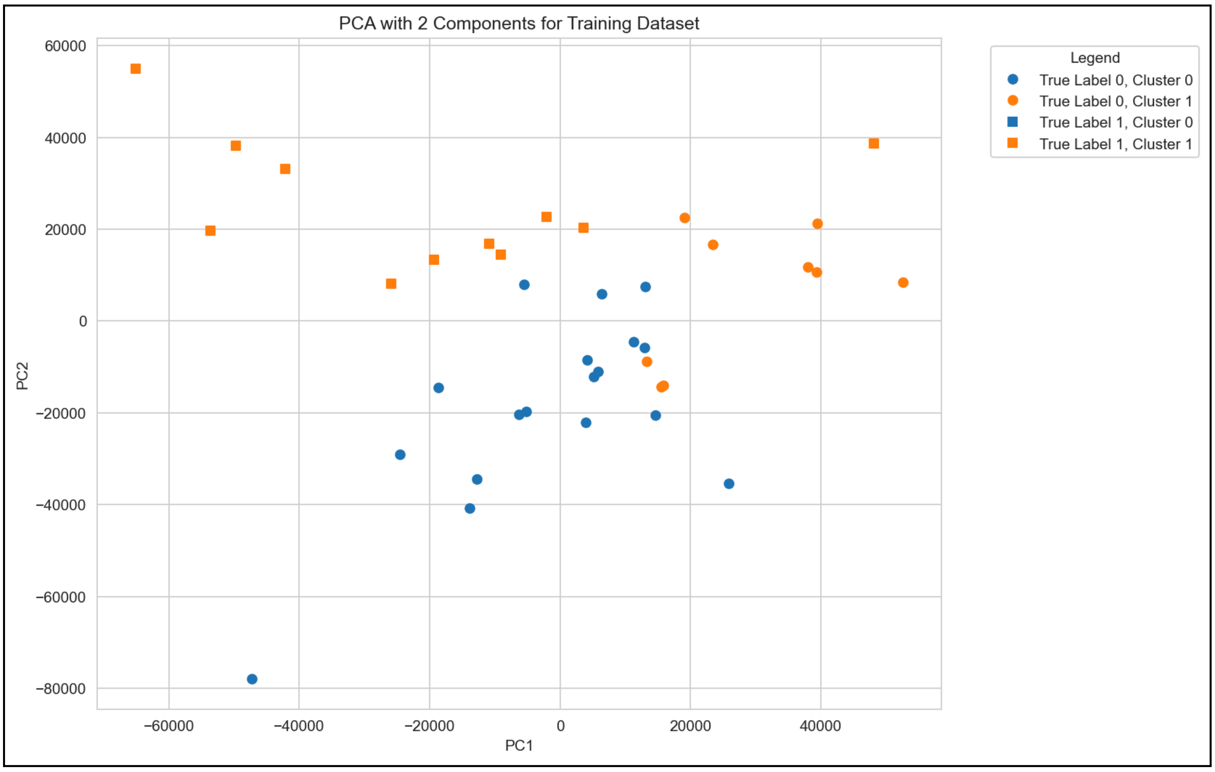
Q3) Without using the class labels in the data, apply an unsupervised technique such as clustering to group the data into two clusters. Report the purity of the best clustering you got (you will need to use the class labels to compute the purity though). Explain the use of this kind of unsupervised clustering in the case of analyzing gene expression data such as this.

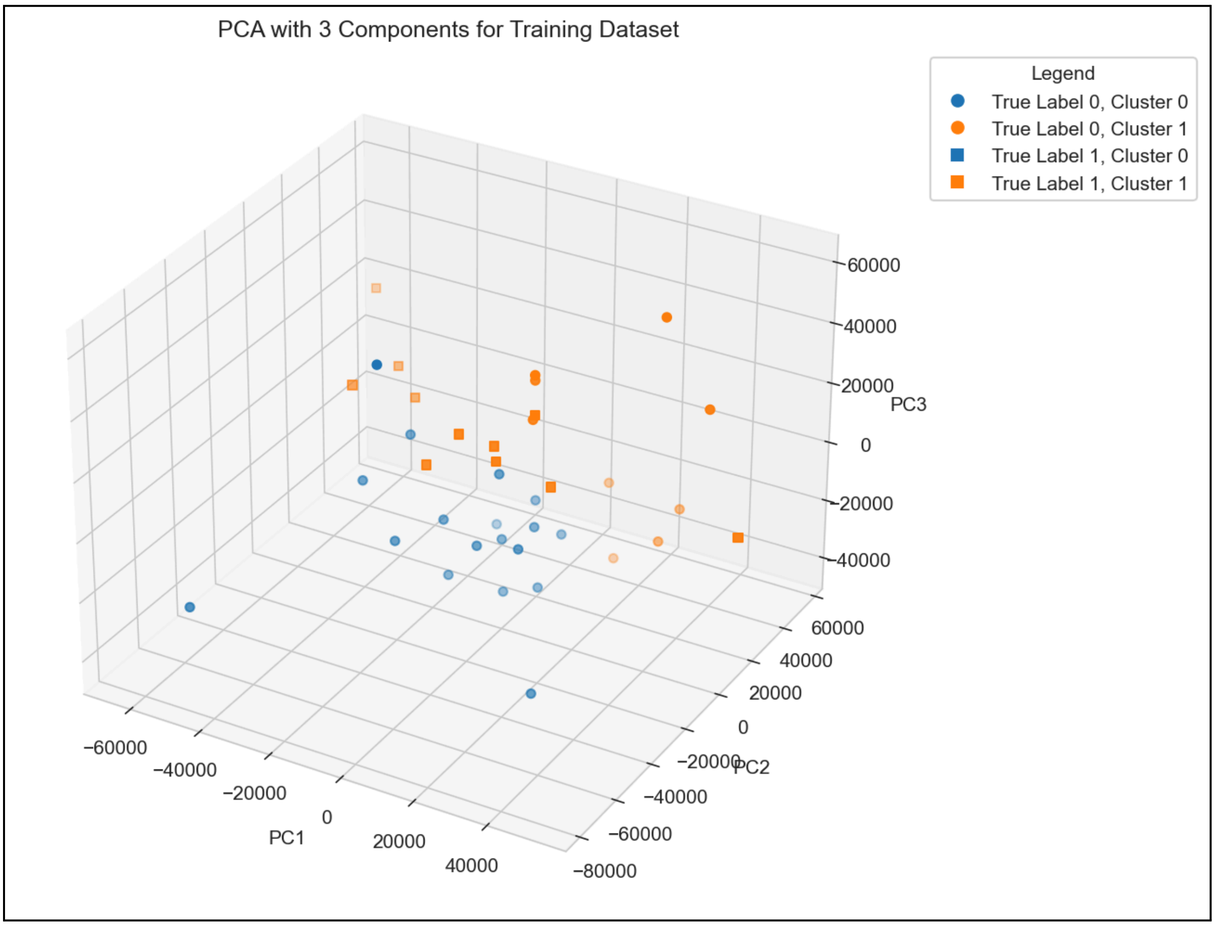
1. Model: k-means clustering
2. Training F1-score and accuracy score: 0.7097 and 0.7632 respectively
3. On training dataset:



```
              precision    recall  f1-score   support

           0       1.00      0.67      0.80        27
           1       0.55      1.00      0.71        11

    accuracy                           0.76        38
   macro avg       0.78      0.83      0.75        38
weighted avg       0.87      0.76      0.77        38
```

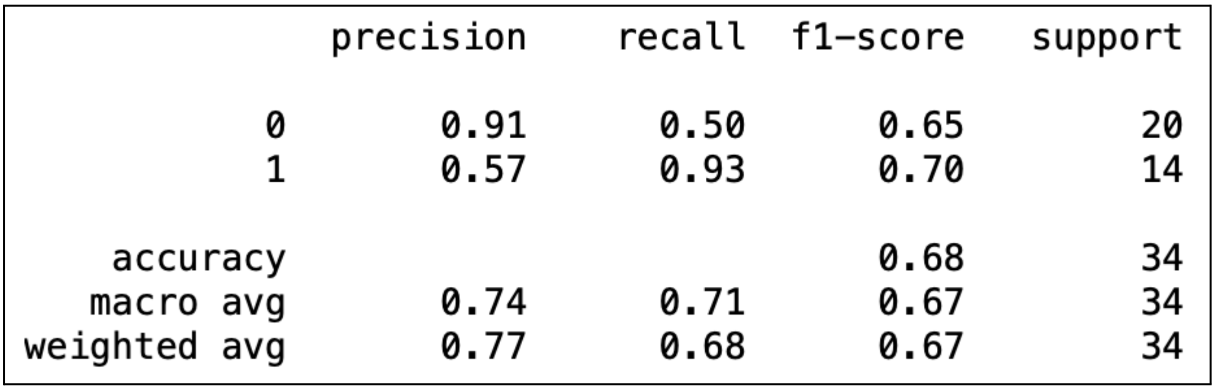



PCA with 2 Components for Training Dataset
PC1
PC2
60000
40000
20000
0
−20000
−40000
−60000
−80000
−60000
−40000
−20000
0
20000
40000
Legend
● True Label 0, Cluster 0
● True Label 0, Cluster 1
■ True Label 1, Cluster 0
■ True Label 1, Cluster 1

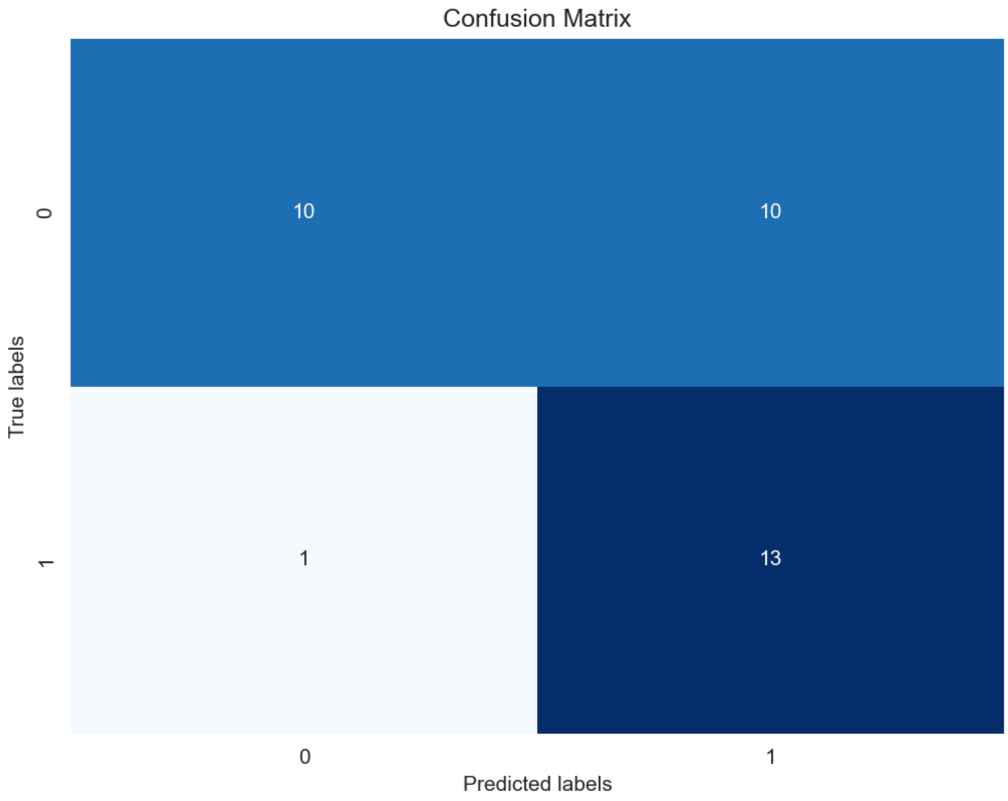

PCA with 3 Components for Training Dataset
Legend
● True Label 0, Cluster 0
● True Label 0, Cluster 1
■ True Label 1, Cluster 0
■ True Label 1, Cluster 1
PC1
PC2
PC3
−60000
−40000
−20000
0
20000
40000
60000
40000
20000
0
−20000
−40000
−60000
−80000
60000
40000
20000
0
−20000
−40000

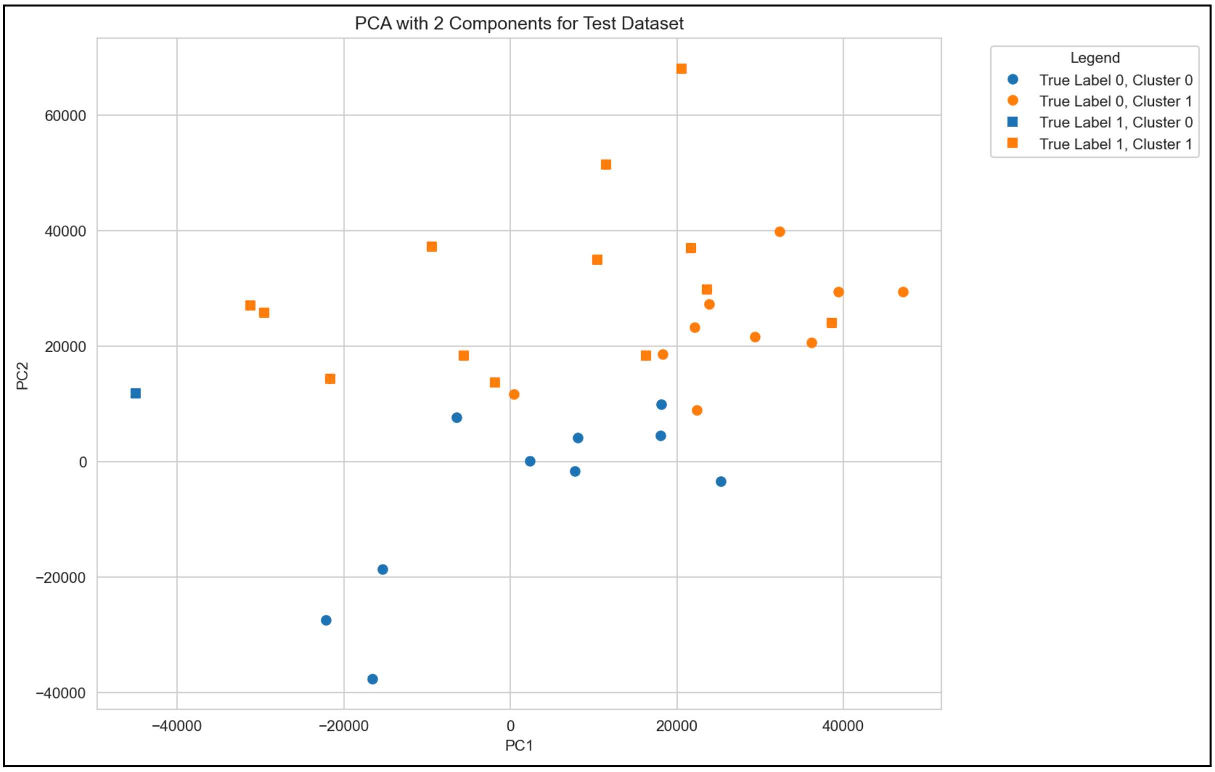
4. Testing F1-score and accuracy score: 0.7027 and 0.6765 respectively

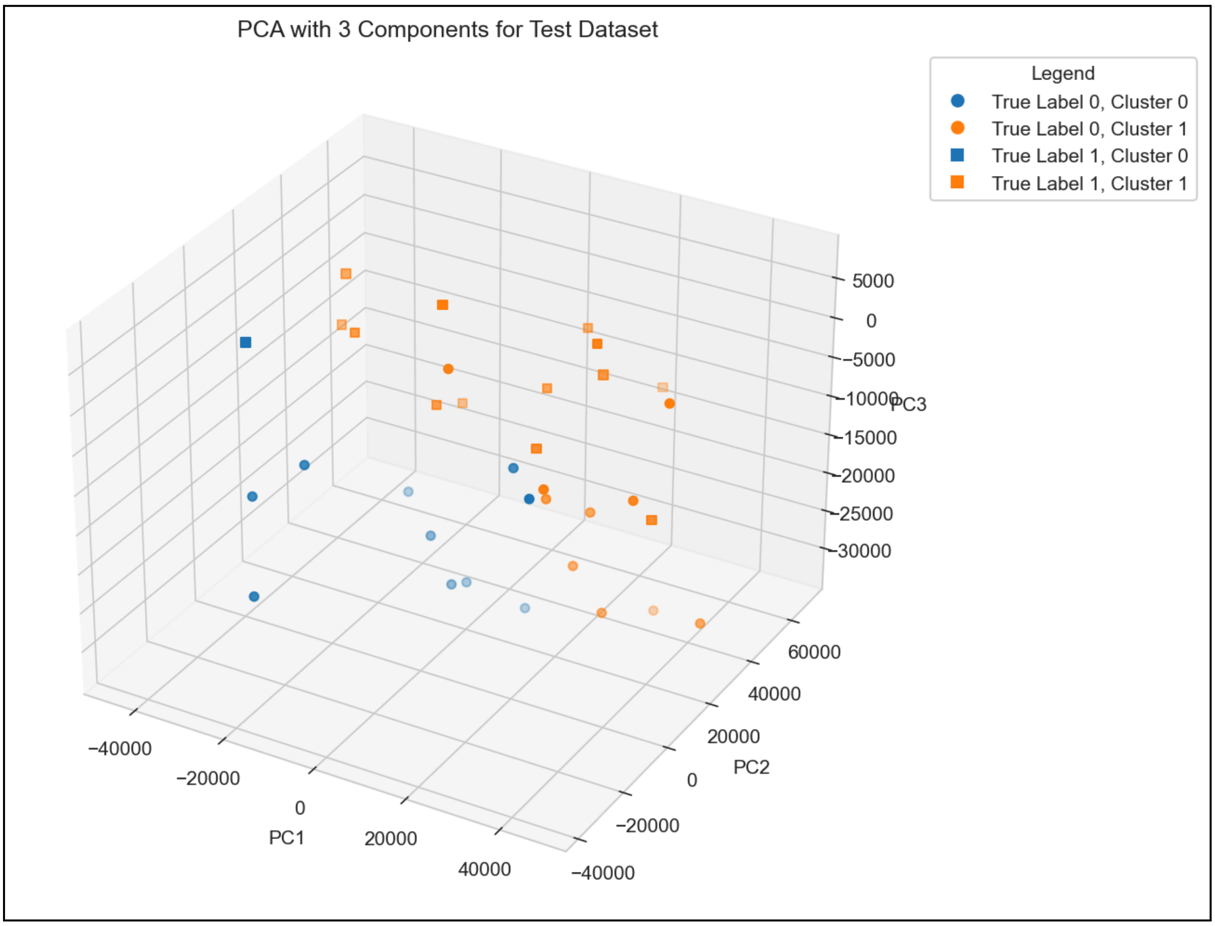
5. On testing dataset:



```
              precision    recall  f1-score   support

           0       0.91      0.50      0.65        20
           1       0.57      0.93      0.70        14

    accuracy                           0.68        34
   macro avg       0.74      0.71      0.67        34
weighted avg       0.77      0.68      0.67        34
```




PCA with 2 Components for Test Dataset
PC1
PC2
−40000
−20000
0
20000
40000
60000
Legend
● True Label 0, Cluster 0
● True Label 0, Cluster 1
■ True Label 1, Cluster 0
■ True Label 1, Cluster 1


PCA with 3 Components for Test Dataset
PC1
PC2
PC3
−40000
−20000
0
20000
40000
60000
5000
−5000
−10000
−15000
−20000
−25000
−30000
Legend
● True Label 0, Cluster 0
● True Label 0, Cluster 1
■ True Label 1, Cluster 0
■ True Label 1, Cluster 1

Unsupervised clustering methods such as k-means can group gene expression data into clusters based on similarities, without using predefined class labels. To assess the quality of clustering, we can measure its purity using known class labels. Purity indicates how well the clusters align with actual classes like AML and ALL, revealing underlying biological patterns. Although supervised learning offers greater accuracy, unsupervised clustering is valuable in exploring new gene interactions and uncovering novel biological insights from gene expression data.