Task 1

a. Initial Model Evaluation:

I started by loading the "glass.data" file and identified the target variable (y) as "Type."

I focused on accuracy and precision as evaluation metrics for classification models Meanwhile studying https://archive.ics.uci.edu/dataset/42/glass+identification , different classifiers were used in articles with calculation Accuracy and Precision , the classifiers like Xgboost , Support vector , Random Forest , Neural Network and logistic regression..

I tested Random Forest and Support Vector Machine (SVM) classifiers with different parameters and standard scaling.

Results:

- Random Forest achieved an accuracy of 1.00, suggesting potential overfitting.
- SVM achieved an accuracy of 0.95.

Best Parameters:

- Random Forest: {'clf__max_depth': 20, 'clf__n_estimators': 50}
- SVM: {'clf__C': 10, 'clf__kernel': 'linear'}

I further tested Logistic Regression, which achieved an accuracy of 0.98, outperforming Random Forest. Best Parameters for Logistic Regression: {'clf__C': 10, 'clf__penalty': 'l2'}.

b. Cross-Validation:

To assess model performance and generalization, I applied different cross-validation techniques, including K-Fold, Stratified K-Fold, and Leave-One-Out.

Results:

- Logistic Regression (K-Fold): Mean Accuracy = 0.90
- Logistic Regression (Stratified K-Fold): Mean Accuracy = 0.90
- Logistic Regression (Leave-One-Out): Mean Accuracy = 0.91
- SVM (K-Fold): Mean Accuracy = 0.85
- SVM (Stratified K-Fold): Mean Accuracy = 0.86
- SVM (Leave-One-Out): Mean Accuracy = 0.88

Based on cross-validation results, Logistic Regression consistently outperformed SVM.

c. AutoML (TPOT) Evaluation:

I applied the TPOTClassifier, an AutoML tool, to optimize the model automatically.

TPOT identified the best pipeline, which was a RandomForestClassifier with specific hyperparameters.

The resulting test accuracy was 0.98, confirming the effectiveness of TPOT in parameter selection.

In summary, the analysis revealed that Logistic Regression consistently performed well across different evaluation methods, outperforming Random Forest and SVM. Additionally, using AutoML (TPOT) helped in finding an optimal model configuration on Random Foret Classifier, emphasizing the importance of parameter selection in achieving the best model performance.

Best pipeline: RandomForestClassifier(input_matrix, bootstrap=True, criterion=gini, max_features=0.9500000000000001, min_samples_leaf=3, min_samples_split=6, n_estimators=100)

AutoML (TPOT) Test Accuracy: 0.98

Task 2

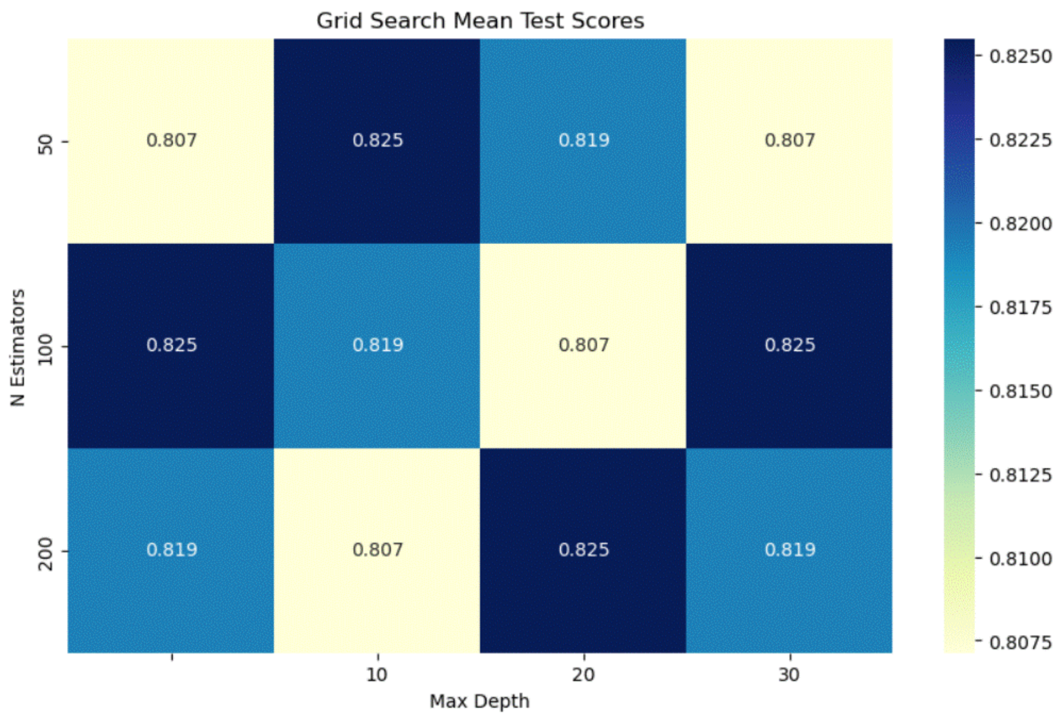
a. First, I loaded the "sonar.all-data.csv" dataset. I mapped the 'M' and 'R' labels to 0 and 1, creating a binary classification task. I then split the data into training (X, y) and validation (X_val, y_val) sets.

Using a Random Forest classification model with specific hyperparameters and StratifiedKFold cross-validation, I obtained the following results:

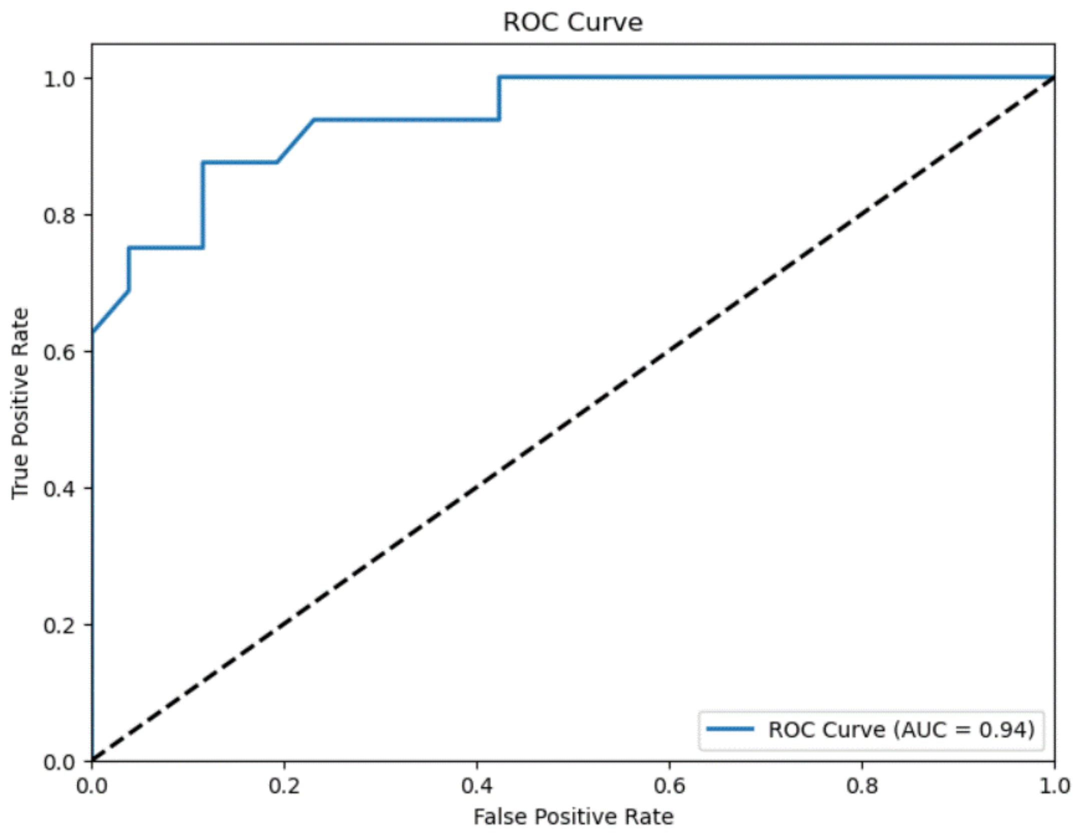
- Best Parameters: {'max_depth': None, 'n_estimators': 100}
- Best Estimator: RandomForestClassifier(random_state=42)
- Validation Accuracy: 0.833 (or 83.3%)

b. After the plotting heatmap, the region with the highest mean test scores, which indicates the best hyperparameter combination and how the performance changes with different values of n_estimators and max_depth. There is any clear relationship between these hyperparameters and the model's accuracy.

It is changed with 2 parameters in navy blue with 0.825 for scores.



c. also plotted a ROC curve to evaluate the model's binary classification performance.



AUC (Area Under the Curve):

- The AUC value represents the overall performance of the model.
- A perfect classifier has an AUC of 1, indicating that it achieves 100% TPR and 0% FPR.
- ROC curves are useful for comparing different models. A model with a higher AUC is generally considered better at classification.

So, AUC 0,94 shows the outperforming Random Forest.

Task 3

Clustering the "20 Newsgroups" dataset into 20 clusters is a challenging task because the dataset is not inherently divided into 20 predefined clusters. The dataset is a collection of newsgroup documents from various topics, but it doesn't come with pre-defined cluster labels.

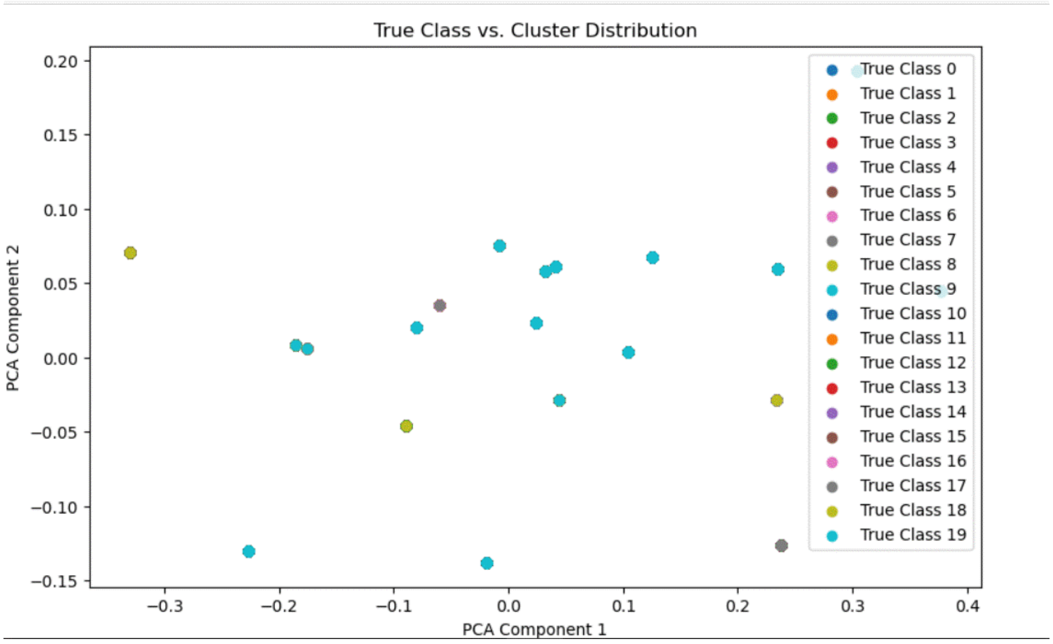
To cluster the data and evaluate the results, I can use unsupervised clustering techniques like K-means clustering. However, without ground truth labels, I won't be able to directly measure the accuracy of the clustering in terms of recovering the original clusters.

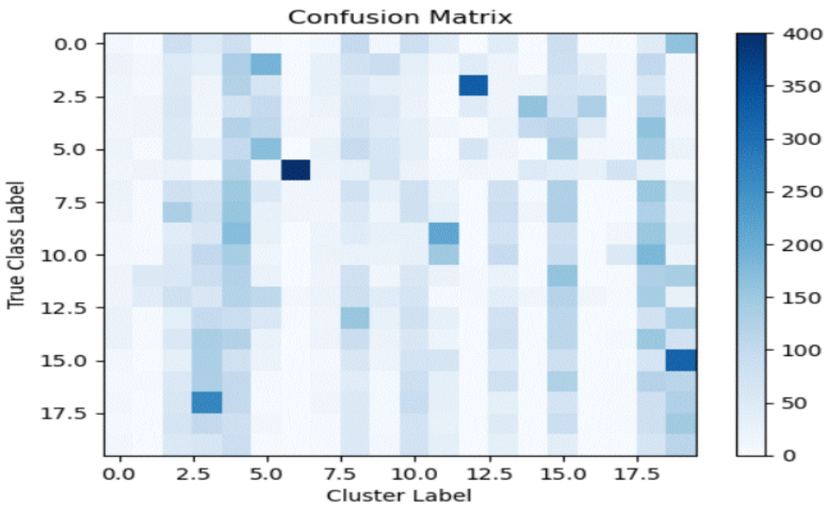
In this example, I loaded the "20 Newsgroups" dataset, concatenate train and test data, with removing headers, footers, and quotes because the removal of headers, footers, and quotes from the text is a common preprocessing step when working with the "20 Newsgroups" dataset and similar text datasets. I used a TF-IDF vectorizer to convert the text data into numerical features and then apply K-means clustering to divide the documents into 20 clusters.

However, I keep in mind that these clusters are unsupervised and may not correspond to the original newsgroup categories.

In practice, recovering the original clusters in this dataset is a complex problem because the dataset was not designed for this purpose, and the categories are quite diverse. The clusters I obtain will likely be based on the patterns in the text data, but they may not correspond to the newsgroup categories in a meaningful way.

This matter is shown in the plots as below.


True Class vs. Cluster Distribution
PCA Component 1
PCA Component 2
True Class 0
True Class 1
True Class 2
True Class 3
True Class 4
True Class 5
True Class 6
True Class 7
True Class 8
True Class 9
True Class 10
True Class 11
True Class 12
True Class 13
True Class 14
True Class 15
True Class 16
True Class 17
True Class 18
True Class 19


Confusion Matrix
True Class Label
Cluster Label