Practical 06

Aim- To implement Graph based clustering and CART algorithm

Theory-

Clustering is an unsupervised learning method that organizes similar data points into groups (clusters) without relying on predefined labels. One powerful variation is graph-based clustering, where data points are modeled as nodes in a graph, and the similarity or distance between them is represented as edge weights. Measures such as cosine similarity, Euclidean distance, or correlation are commonly used for this purpose.

Once a similarity graph is built, different algorithms can be applied to identify clusters, including:

- Minimum Spanning Tree (MST) clustering
- Spectral Clustering (using the graph Laplacian to project data into a lower-dimensional space, followed by k-means)
- Community Detection techniques

Graph-based clustering is especially effective for complex, non-convex datasets that traditional methods like k-means cannot separate well.

CART (Classification and Regression Trees)
The CART algorithm, introduced by Breiman et al. in 1986, is one of the most widely used decision tree techniques. Unlike clustering, CART is a supervised learning method that can be applied to both:

- Classification problems (predicting categorical labels)
- Regression problems (predicting continuous values)

CART uses the Gini Index for classification tasks and Mean Squared Error (MSE) for regression tasks. It works by recursively splitting the dataset based on the feature and threshold that provides the best separation. The resulting tree is always binary, meaning each split produces exactly two branches.

Key strengths of CART include:

- Simplicity and interpretability
- Ability to handle both numerical and categorical features

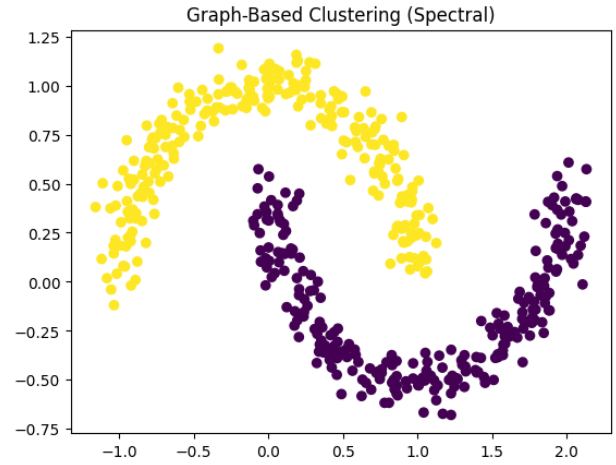
However, CART is prone to overfitting if the tree grows too deep. To address this, methods like pruning, depth constraints, or ensemble models (e.g., Random Forests) are commonly used.

Code-

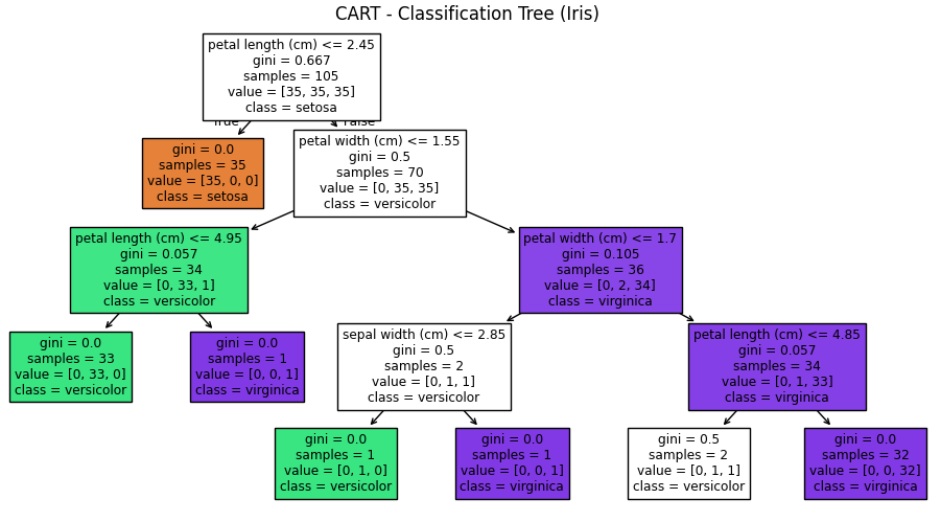
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons, load_iris, make_regression
from sklearn.cluster import SpectralClustering
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor,
plot_tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, mean_squared_error,
r2_score

RANDOM_STATE = 42

def graph_based_clustering_demo():
    X, _ = make_moons(n_samples=500, noise=0.08,
random_state=RANDOM_STATE)
    sc = SpectralClustering(n_clusters=2, affinity="nearest_neighbors",
n_neighbors=10, random_state=RANDOM_STATE)
    labels = sc.fit_predict(X)
    plt.scatter(X[:, 0], X[:, 1], c=labels, cmap="viridis")
    plt.title("Graph-Based Clustering (Spectral)")
    plt.show()

def cart_classification_demo():
    iris = load_iris()
    X_train, X_test, y_train, y_test = train_test_split(iris.data,
iris.target, test_size=0.3, random_state=RANDOM_STATE,
stratify=iris.target)
    clf = DecisionTreeClassifier(criterion="gini", max_depth=4,
random_state=RANDOM_STATE)
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print(f"CART Classification Accuracy (Iris): {acc:.3f}")
    plt.figure(figsize=(12, 6))
    plot_tree(clf, feature_names=iris.feature_names,
class_names=iris.target_names, filled=True)
    plt.title("CART - Classification Tree (Iris)")
    plt.show()

def cart_regression_demo():
    X, y = make_regression(n_samples=500, n_features=1, noise=15,
random_state=RANDOM_STATE)
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=RANDOM_STATE)
    reg = DecisionTreeRegressor(criterion="squared_error", max_depth=4,
random_state=RANDOM_STATE)
    reg.fit(X_train, y_train)
```

```
    y_pred = reg.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    print(f"CART Regression -> MSE: {mse:.2f}, R²: {r2:.3f}")
    X_line = np.linspace(X.min(), X.max(), 500).reshape(-1, 1)
    y_line = reg.predict(X_line)
    plt.scatter(X, y, s=15, label="Data")
    plt.plot(X_line, y_line, color="red", label="CART Prediction")
    plt.title("CART - Regression Tree")
    plt.legend()
    plt.show()

if __name__ == "__main__":
    graph_based_clustering_demo()
    cart_classification_demo()
    cart_regression_demo()
```

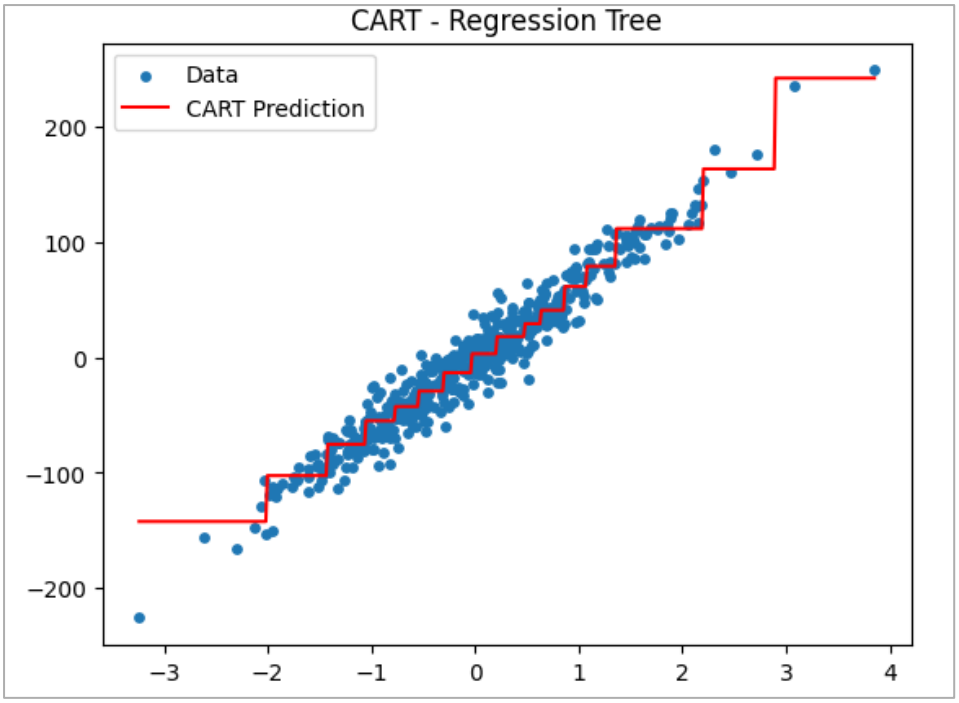
Output-




```
CART Classification Accuracy (Iris): 0.889
```




```
CART Regression -> MSE: 318.60, R²: 0.917
```



Conclusion- In this practical, we implemented Graph-based Clustering and the CART algorithm, covering both unsupervised and supervised machine learning paradigms. Graph-based clustering was shown to effectively group data with non-linear structures, while CART provided an interpretable decision-making model for classification. Together, these methods demonstrated the importance of clustering for pattern discovery and decision trees for prediction, with strong relevance to real-world applications such as fraud detection and anomaly detection in blockchain systems.