DAYANANDA SAGAR ACADEMY OF TECHNOLOGY & MANAGEMENT

Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore- 560082

iliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi

CSE,ISE,ECE,EEE,ME,CE Branches Accredited by NBA, New Delhi

NAAC Accredited with A+ Grade

Department of Artificial Intelligence and Machine Learning

Academic Year 2023-24

DATA SCIENCE AND ITS APPLICATIONS LABORATORY (21AD62)

CONTENTS

DATA SCIENCE AND ITS APPLICATIONS LABORATORY

Course Code	21AD62	CIE Marks	**50**
Teaching Hours/Weeks (L: T: P: S)	3:0:2:0	SEE Marks	**50**
Total Hours of Pedagogy	40T + 20P	Total Marks	100
Credits	04	Exam Hours	03

Course Learning Objectives:

CLO 1. Demonstrate the proficiency with statistical analysis of data to derive insight from results and interpret the data findings visually

CLO 2. Utilize the skills in data management by obtaining, cleaning and transforming the data

CLO 3. Make use of machine learning models to solve the business-related challenges

CLO 4. Experiment with decision trees, neural network layers and data partition.

CLO5. Demonstrate how social clustering shape individuals and groups in contemporary society.

Module 1
Installation of Python/R language, Visual Studio code editors can be demonstrated along with Kaggle data set usage. 2. Write programs in Python/R and Execute them in either Visual Studio Code or PyCharm Community Edition or any other suitable environment. 3. A study was conducted to understand the effect of number of hours the students spent studying on their performance in the final exams. Write a code to plot line chart with number of hours spent studying on x-axis and score in final exam on y-axis. Use a red '*' as the point character, label the axes and give the plot a title. Number of hrs spent studying (x):10 9 2 15 10 16 11 16 Score inthe finalexam (0– 100)(y): 95 80 10 50 45 98 38 93 4. For the given dataset mtcars.csv (www.kaggle.com/ruiromanini/mtcars), plot a histogram to check the frequency distribution of the variable 'mpg' (Miles per gallon)
Module 2
1. Consider the books dataset BL-Flickr-Images-Book.csv from Kaggle (https://www.kaggle.com/adeyoyintemidayo/publication-of-books) which contains information about books. Write a program to demonstrate the following. • Import the data into a DataFrame • Find and drop the columns which are irrelevant for the book information. • Change the Index of the DataFrame • Tidy up fields in the data such as date of publication with the help of simple regular expression. • Combine str methods with NumPy to clean columns
Module 3
1. Train a regularized logistic regression classifier on the iris dataset (https://archive.ics.uci.edu/ml/machine-learning-databases/iris/ or the inbuilt iris dataset) using sklearn. Train the model with the following hyperparameter C = 1e4 and report the best classification accuracy. 2. Train an SVM classifier on the iris dataset using sklearn. Try different kernels and the associated hyperparameters. Train model with the following set of hyperparameters RBFkernel, gamma=0.5, one-vs-rest classifier, no-feature-normalization. Also try C=0.01,1,10C=0.01,1,10. For the above set of hyperparameters, find the best classification accuracy along with total number of support vectors on the test data
Module 4
1.Consider the following dataset. Write a program to demonstrate the working of the decision tree

based ID3 algorithm.
2. Consider the dataset spiral.txt (https://bit.ly/2Lm75Ly). The first two columns in the dataset corresponds to the co-ordinates of each data point. The third column corresponds to the actual cluster label. Compute the rand index for the following methods:
- K – means Clustering
- Single – link Hierarchical Clustering
- Complete link hierarchical clustering.
- Also visualize the dataset and which algorithm will be able to recover the true clusters.

Module 5
Mini Project – Simple web scrapping in social media

Course Learning Objectives:

CO 1: Identify and demonstrate data using visualization tools

CO 2: Make use of Statistical hypothesis tests to choose the properties of data, curate and manipulate data.
CO 3: Utilize the skills of machine learning algorithms and techniques and develop models.

CO 4: Demonstrate the construction of decision tree and data partition using clustering.

CO5: Experiment with social network analysis and make use of natural language processing skills to develop data driven applications.

Assessment Details

Practical Sessions need to be assessed by appropriate rubrics and viva-voce method. This will contribute to **20 marks**.

Note: Minimum of 80% of the laboratory components have to be covered.

- Rubrics for each Experiment taken average for all Lab components– 15 Marks.
- Viva-Voce– 5 Marks (more emphasized on demonstration topics)

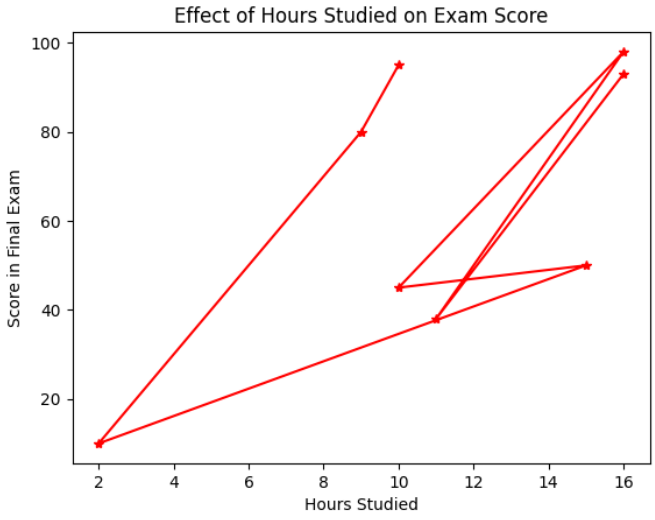
Program 1

A study was conducted to understand the effect of number of hours the students spent studying on their performance in the final exams. Write a code to plot line chart with number of hours spent studying on x-axis and score in final exam on y-axis. Use a red '*' as the point character, label the axes and give the plot a title.import matplotlib.pyplot as plt

```
# Sample data
hours_studied = [10, 9, 2, 15, 10, 16, 11, 16]
exam_scores = [95, 80, 10, 50, 45, 98, 38, 93]
# Plot line chart
plt.plot(hours_studied, exam_scores, marker='*', color='red', linestyle='-')
# Add labels and title
plt.xlabel('Hours Studied')
plt.ylabel('Score in Final Exam')
plt.title('Effect of Hours Studied on Exam Score')

# Show plot
plt.show()
```

Output

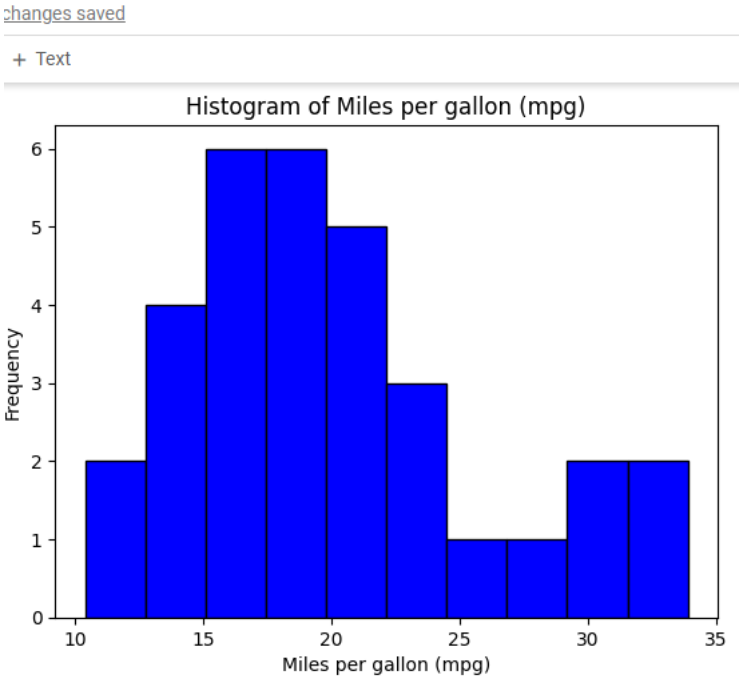


Program 2

For the given dataset mtcars.csv (www.kaggle.com/ruiromanini/mtcars), plot a histogram to check the frequency distribution of the variable ‘mpg’ (Miles per gallon)

```
import pandas as pd
import matplotlib.pyplot as plt
# Load the dataset
mtcars = pd.read_csv("mtcars.csv")
# Plot histogram
plt.hist(mtcars['mpg'], bins=10, color='blue', edgecolor='black')
# Add labels and title
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('Frequency')
plt.title('Histogram of Miles per gallon (mpg)')
# Show plot
plt.show()
```

Output



Program 3

Consider the books dataset BL-Flickr-Images-Book.csv from Kaggle (https://www.kaggle.com/adeyoyintemidayo/publication-of-books) which contains information about books. Write a program to demonstrate the following.

- Import the data into a DataFrame
- Find and drop the columns which are irrelevant for the book information.
- Change the Index of the DataFrame
- Tidy up fields in the data such as date of publication with the help of simple regular expression.
- Combine str methods with NumPy to clean columns

```
import pandas as pd
import numpy as np

# Import the data into a DataFrame
books_df = pd.read_csv("BL-Flickr-Images-Book.csv")

# Display the first few rows of the DataFrame
print("Original DataFrame:")
print(books_df.head())

# Find and drop the columns which are irrelevant for the book information
irrelevant_columns = ['Edition Statement', 'Corporate Author', 'Corporate Contributors', 'Former owner', 'Engraver', 'Contributors', 'Issuance type', 'Shelfmarks']
books_df.drop(columns=irrelevant_columns, inplace=True)

# Change the Index of the DataFrame
books_df.set_index('Identifier', inplace=True)

# Tidy up fields in the data such as date of publication with the help of simple regular expression
books_df['Date of Publication'] = books_df['Date of Publication'].str.extract(r'^(\d{4})', expand=False)

# Combine str methods with NumPy to clean columns
books_df['Date of Publication'] = pd.to_numeric(books_df['Date of Publication'], errors='coerce')

# Display the cleaned DataFrame
print("\nCleaned DataFrame:")
print(books_df.head())
```

Output:

```
Original DataFrame:
   Identifier             Edition Statement      Place of Publication  \
0         206                           NaN                    London
1         216                           NaN  London; Virtue & Yorston
2         218                           NaN                    London
3         472                           NaN                    London
4         480  A new edition, revised, etc.                    London

  Date of Publication             Publisher  \
0         1879 [1878]      S. Tinsley & Co.
1                1868          Virtue & Co.
2                1869  Bradbury, Evans & Co.
3                1851         James Darling
4                1857   Wertheim & Macintosh

                                          Title     Author  \
0                 Walter Forbes. [A novel.] By A. A      A. A.
1  All for Greed. [A novel. The dedication signed...  A., A. A.
2  Love the Avenger. By the author of "All for Gr...  A., A. A.
3  Welsh Sketches, chiefly ecclesiastical, to the...  A., E. S.
4  [The World in which I live, and my place in it...  A., E. S.

                              Contributors  Corporate Author  \
0                            FORBES, Walter.               NaN
1  BLAZE DE BURY, Marie Pauline Rose - Baroness               NaN
2  BLAZE DE BURY, Marie Pauline Rose - Baroness               NaN
3                 Appleyard, Ernest Silvanus.               NaN
4                       BROOME, John Henry.                 NaN

   Corporate Contributors Former owner  Engraver Issuance type  \
0                     NaN          NaN       NaN   monographic
1                     NaN          NaN       NaN   monographic
2                     NaN          NaN       NaN   monographic
3                     NaN          NaN       NaN   monographic
4                     NaN          NaN       NaN   monographic

                                      Flickr URL  \
0  http://www.flickr.com/photos/britishlibrary/ta...
1  http://www.flickr.com/photos/britishlibrary/ta...
2  http://www.flickr.com/photos/britishlibrary/ta...
3  http://www.flickr.com/photos/britishlibrary/ta...
4  http://www.flickr.com/photos/britishlibrary/ta...

                        Shelfmarks
0  British Library HMNTS 12641.b.30.
1  British Library HMNTS 12626.cc.2.
2  British Library HMNTS 12625.dd.1.
3  British Library HMNTS 10369.bbb.15.
4   British Library HMNTS 9007.d.28.

Cleaned DataFrame:
            Place of Publication  Date of Publication  \
Identifier
206                       London               1879.0
216       London; Virtue & Yorston               1868.0
218                       London               1869.0
472                       London               1851.0
480                       London               1857.0

                    Publisher  \
Identifier
206            S. Tinsley & Co.
216                Virtue & Co.
218          Bradbury, Evans & Co.
472                 James Darling
480           Wertheim & Macintosh

                                                    Title     Author  \
Identifier
206                         Walter Forbes. [A novel.] By A. A      A. A.
216         All for Greed. [A novel. The dedication signed...  A., A. A.
218         Love the Avenger. By the author of "All for Gr...  A., A. A.
472         Welsh Sketches, chiefly ecclesiastical, to the...  A., E. S.
480         [The World in which I live, and my place in it...  A., E. S.

                                                 Flickr URL
Identifier
206         http://www.flickr.com/photos/britishlibrary/ta...
216         http://www.flickr.com/photos/britishlibrary/ta...
218         http://www.flickr.com/photos/britishlibrary/ta...
472         http://www.flickr.com/photos/britishlibrary/ta...
480         http://www.flickr.com/photos/britishlibrary/ta...
```

Program 4

Train a regularized logistic regression classifier on the iris dataset (https://archive.ics.uci.edu/ml/machine-learning-databases/iris/ or the inbuilt iris dataset) using sklearn. Train the model with the following hyper parameter C = 1e4 and report the best classification accuracy

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the logistic regression classifier with regularization (L2 penalty) and C=1e4
log_reg = LogisticRegression(C=1e4, penalty='l2', solver='lbfgs', max_iter=1000)
log_reg.fit(X_train, y_train)

# Predict on the test set
y_pred = log_reg.predict(X_test)

# Calculate classification accuracy
accuracy = accuracy_score(y_test, y_pred)

print("Classification Accuracy:", accuracy)
```

.**Output:**

Classification Accuracy: 1.0

Program 5

Train an SVM classifier on the iris dataset using sklearn. Try different kernels and the associated hyper parameters. Train model with the following set of hyper parameters RB Fkernel, gamma=0.5, one-vs-rest classifier, no-feature-normalization. Also try C=0.01,1,10C=0.01,1,10. For the above set of hyper parameters, find the best classification accuracy along with total number of support vectors on the test data

```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Define the hyperparameters to try
kernels = ['rbf']
gammas = [0.5]
Cs = [0.01, 1, 10]

best_accuracy = 0
best_parameters = None
best_support_vectors = None

# Iterate over different combinations of hyperparameters
for kernel in kernels:
    for gamma in gammas:
        for C in Cs:
            # Train the SVM classifier
            svm_clf = SVC(kernel=kernel, gamma=gamma, C=C, decision_function_shape='ovr')
            svm_clf.fit(X_train, y_train)

            # Predict on the test set
            y_pred = svm_clf.predict(X_test)

             # Calculate classification accuracy
            accuracy = accuracy_score(y_test, y_pred)
```

```
        # Get the total number of support vectors
        total_support_vectors = np.sum(svm_clf.n_support_)

        print(f"Kernel: {kernel}, Gamma: {gamma}, C: {C}, Accuracy: {accuracy}, Total Support Vectors: {total_support_vectors}")

        # Check if this model has the best accuracy so far
        if accuracy > best_accuracy:
            best_accuracy = accuracy
            best_parameters = (kernel, gamma, C)
            best_support_vectors = total_support_vectors

print("\nBest Classification Accuracy:", best_accuracy)
print("Best Hyperparameters:", best_parameters)
print("Total Support Vectors for Best Model:", best_support_vectors)
```

Output:

```
Kernel: rbf, Gamma: 0.5, C: 0.01, Accuracy: 0.3, Total Support Vectors: 120
Kernel: rbf, Gamma: 0.5, C: 1, Accuracy: 1.0, Total Support Vectors: 39
Kernel: rbf, Gamma: 0.5, C: 10, Accuracy: 1.0, Total Support Vectors: 31

Best Classification Accuracy: 1.0
Best Hyperparameters: ('rbf', 0.5, 1)
Total Support Vectors for Best Model: 39
```

Program 6

Write a program to demonstrate the working of the decision tree based ID3 algorithm.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Define the dataset
data = [
    ['Low', 'Low', 2, 'No', 'Yes'],
    ['Low', 'Med', 4, 'Yes', 'Yes'],
    ['Low', 'Low', 4, 'No', 'Yes'],
    ['Low', 'Med', 4, 'No', 'No'],
    ['Low', 'High', 4, 'No', 'No'],
    ['Med', 'Med', 4, 'No', 'No'],
    ['Med', 'Med', 4, 'Yes', 'Yes'],
    ['Med', 'High', 2, 'Yes', 'No'],
    ['Med', 'High', 5, 'No', 'Yes'],
    ['High', 'Med', 4, 'Yes', 'Yes'],
    ['High', 'Med', 2, 'Yes', 'Yes'],
    ['High', 'High', 2, 'Yes', 'No'],
    ['High', 'High', 5, 'Yes', 'Yes']
]

# Convert data to DataFrame
import pandas as pd
df = pd.DataFrame(data, columns=['Price', 'Maintenance', 'Capacity', 'Airbag', 'Profitable'])

# Encode categorical features
label_encoders = {}
for column in ['Price', 'Maintenance', 'Airbag', 'Profitable']:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# Separate features and target variable
X = df.drop(columns=['Profitable'])
y = df['Profitable']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the decision tree classifier
clf = DecisionTreeClassifier(criterion='entropy')  # ID3 algorithm uses information gain (entropy) for splitting
clf.fit(X_train, y_train)

# Predictions on the test set
```

```
y_pred = clf.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Output:
Classification Accuracy: 1.0

Program 7

Consider the dataset spiral.txt (https://bit.ly/2Lm75Ly). The first two columns in the dataset corresponds to the co-ordinates of each data point. The third column corresponds to the actual cluster label. Compute the rand index for the following methods:

- K – means Clustering
- Single – link Hierarchical Clustering
- Complete link hierarchical clustering.
- Also visualize the dataset and which algorithm will be able to recover the true clusters.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.metrics import adjusted_rand_score

# Load the dataset
data = np.loadtxt("spiral.txt")

# Extract coordinates and true cluster labels
X = data[:, :2]
true_labels = data[:, 2]

# Visualize the dataset
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=true_labels, cmap='viridis', s=50, alpha=0.8)
plt.title("True Clusters")
plt.xlabel("X1")
plt.ylabel("X2")
plt.show()

# K-means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans_pred = kmeans.fit_predict(X)
kmeans_rand_index = adjusted_rand_score(true_labels, kmeans_pred)
print("Rand Index for K-means:", kmeans_rand_index)

# Single-link Hierarchical Clustering
single_link = AgglomerativeClustering(n_clusters=3, linkage='single')
single_link_pred = single_link.fit_predict(X)
single_link_rand_index = adjusted_rand_score(true_labels, single_link_pred)
print("Rand Index for Single-link Hierarchical Clustering:", single_link_rand_index)

# Complete-link Hierarchical Clustering
complete_link = AgglomerativeClustering(n_clusters=3, linkage='complete')
complete_link_pred = complete_link.fit_predict(X)
complete_link_rand_index = adjusted_rand_score(true_labels, complete_link_pred)
print("Rand Index for Complete-link Hierarchical Clustering:", complete_link_rand_index)

# Visualize the clustering results
plt.figure(figsize=(15, 5))
```

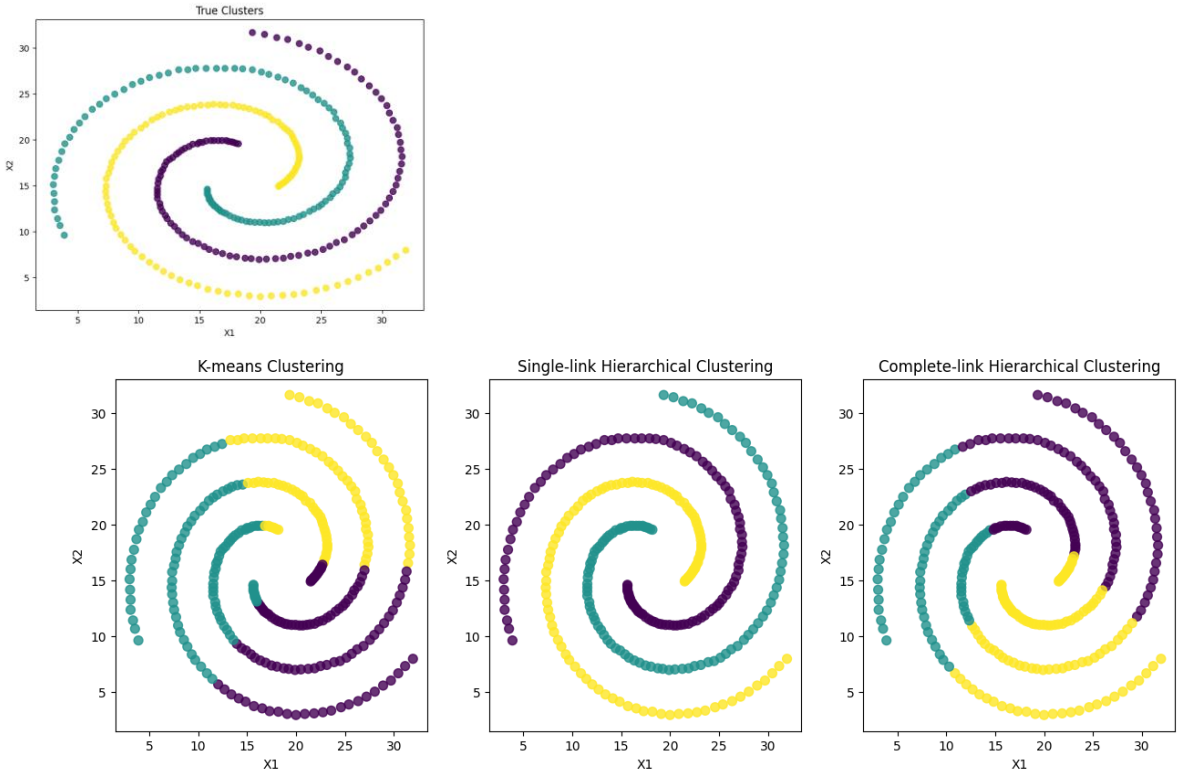
```
plt.subplot(1, 3, 1)
plt.scatter(X[:, 0], X[:, 1], c=kmeans_pred, cmap='viridis', s=50, alpha=0.8)
plt.title("K-means Clustering")
plt.xlabel("X1")
plt.ylabel("X2")

plt.subplot(1, 3, 2)
plt.scatter(X[:, 0], X[:, 1], c=single_link_pred, cmap='viridis', s=50, alpha=0.8)
plt.title("Single-link Hierarchical Clustering")
plt.xlabel("X1")
plt.ylabel("X2")

plt.subplot(1, 3, 3)
plt.scatter(X[:, 0], X[:, 1], c=complete_link_pred, cmap='viridis', s=50, alpha=0.8)
plt.title("Complete-link Hierarchical Clustering")
plt.xlabel("X1")
plt.ylabel("X2")

plt.show()
```

Output:



Viva-voce Questions:

1. What is data science, and why is it important?
2. Explain the CRISP-DM methodology.
3. What are the key steps involved in the data science process?
4. What is the difference between supervised and unsupervised learning?
5. Can you give examples of supervised and unsupervised learning algorithms?
6. What is overfitting, and how can it be avoided?
7. Describe the bias-variance tradeoff.
8. What are some common techniques for feature selection?
9. How does regularization help in machine learning models?
10. What is cross-validation, and why is it used?
11. Explain the concept of ensemble learning.
12. What is the difference between bagging and boosting?
13. How do decision trees work, and what are their advantages and disadvantages?
14. What are some techniques for handling missing data?
15. Explain the difference between correlation and causation.
16. What is dimensionality reduction, and why is it important?
17. Describe principal component analysis (PCA).
18. How does PCA help in dimensionality reduction?
19. What are the assumptions of linear regression?
20. How do you interpret the coefficients in linear regression?
21. What is logistic regression, and when is it used?
22. Explain the confusion matrix and its components.
23. What are precision and recall, and how are they calculated?
24. What is the F1 score, and why is it useful?
25. Describe the ROC curve and AUC.
26. What is clustering, and what are some common clustering algorithms?
27. Explain K-means clustering.
28. What is the elbow method used for in K-means clustering?
29. Describe hierarchical clustering.
30. What is the difference between K-means and hierarchical clustering?
31. What is anomaly detection, and how is it useful?
32. Explain the use of Gaussian mixture models (GMM) in anomaly detection.
33. What is the purpose of time series analysis?
34. Describe the components of a time series.
35. What are some common techniques for time series forecasting?
36. Explain ARIMA models.
37. What is the purpose of sentiment analysis?
38. How do you preprocess text data for sentiment analysis?
39. Describe the bag-of-words model.
40. What is the purpose of word embeddings in natural language processing?
41. Explain the difference between word2vec and GloVe.
42. What is topic modeling, and how is it useful?
43. Describe Latent Dirichlet Allocation (LDA).
44. How do you evaluate a machine learning model?
45. What is the purpose of hyperparameter tuning?
46. Explain grid search and random search for hyperparameter tuning.
47. What is the bias of a machine learning model, and how can it be reduced?
48. How do you handle imbalanced datasets?
49. What is the curse of dimensionality, and how does it affect machine learning algorithms?
50. How do you deploy a machine learning model into production?