DATA SCIENCE AND ITS APPLICATIONS – 21AD62

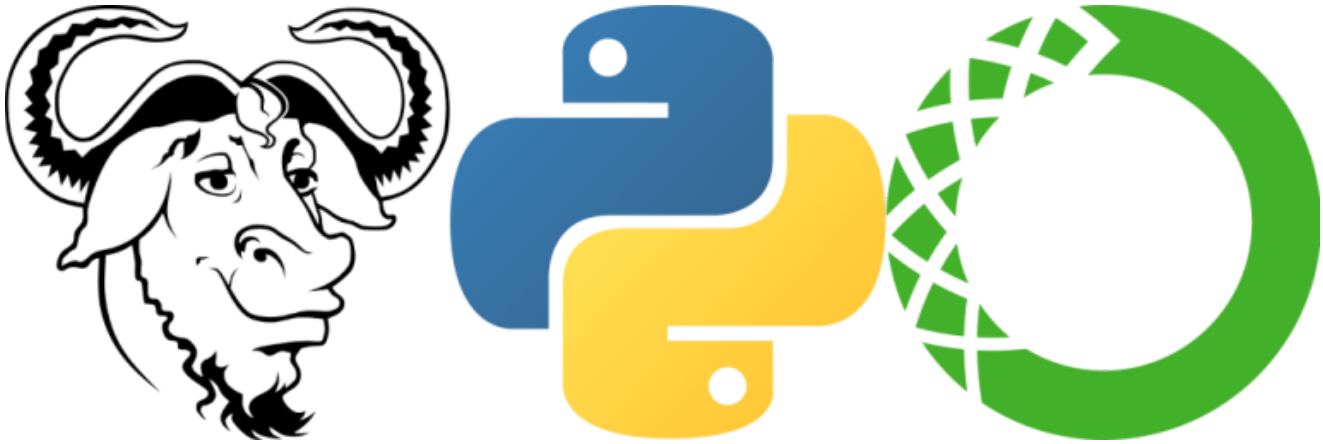
Prabodh C P 3 days ago

In this blog post, you will find solutions for the **Data Science And Its Applications (21AD62)** course work for the VI semester of **VTU** university. To follow along, you will need to set up a Python programming environment. We recommend using the Anaconda Python Distribution with Jupyter Notebook/ Spyder as the integrated development environment (IDE). You can find the lab syllabus on the university's website or here.

Syllabus

21AD62 Download

For detailed instructions on setting up the Python programming environment on Ubuntu, please refer to my previous blog, which can be found below.

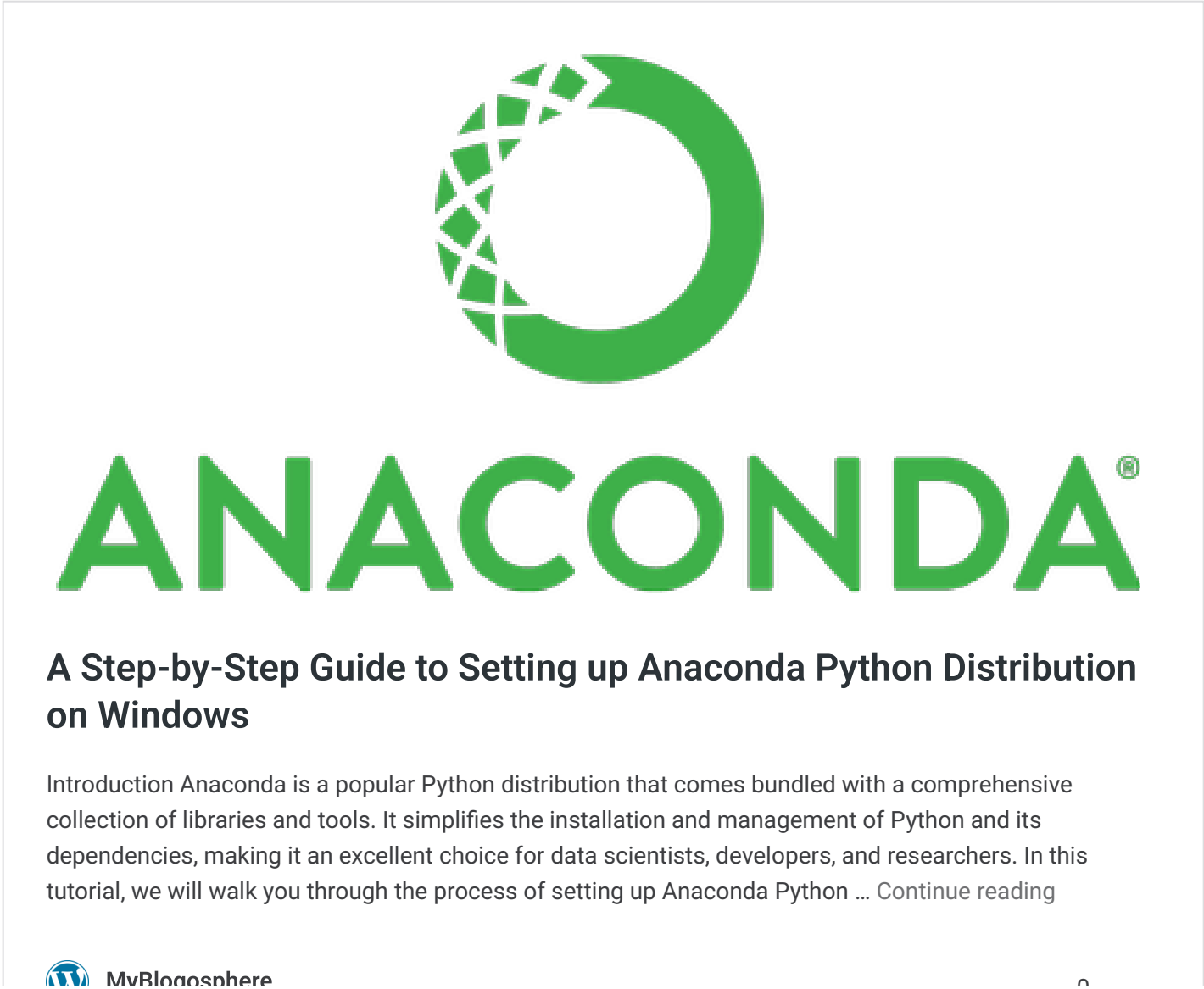


Setting up Anaconda Python Programming Environment on Ubuntu

This tutorial will help you in setting up a Anaconda Python 3 programming environment on Ubuntu 22.04. (Should also work on previous release Ubuntu 20.04 as well). If you do not have Ubuntu 22.04 installed on your machine, you can follow the detailed description of installing Ubuntu 22.04 Desktop in the following link below. https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview … Continue reading

MyBlogosphere 0

If you are looking for step-by-step instructions on how to set up the Python programming environment on a Windows system, I have provided detailed guidance in my previous blog. You can access the blog below for all the information you need.



After getting the necessary development environment setup, Now lets focus on the solutions.

1. Module 1
   a. Students performance in the final exams
   b. Histogram to check the frequency distribution

2. Module 2
   a. Kaggle Book Data set

3. Module 3
   a. Logistic Regression

b. SVM classifier
4. Module 4
   a. Decision Tree based ID3 algorithm
   b. Clustering
5. Module 5
   a. Mini Project

Module 1

Question 3

Students performance in the final exams

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 in the final exam (0 – 100) (y)	95	80	10	50	45	98	38	93

Python Code

```
import matplotlib.pyplot as plt

hours = [10,9,2,15,10,16,11,16]
score = [95,80,10,50,45,98,38,93]

# Plotting the line chart
plt.plot(hours, score, marker='*', color='red', linestyle='-')
```

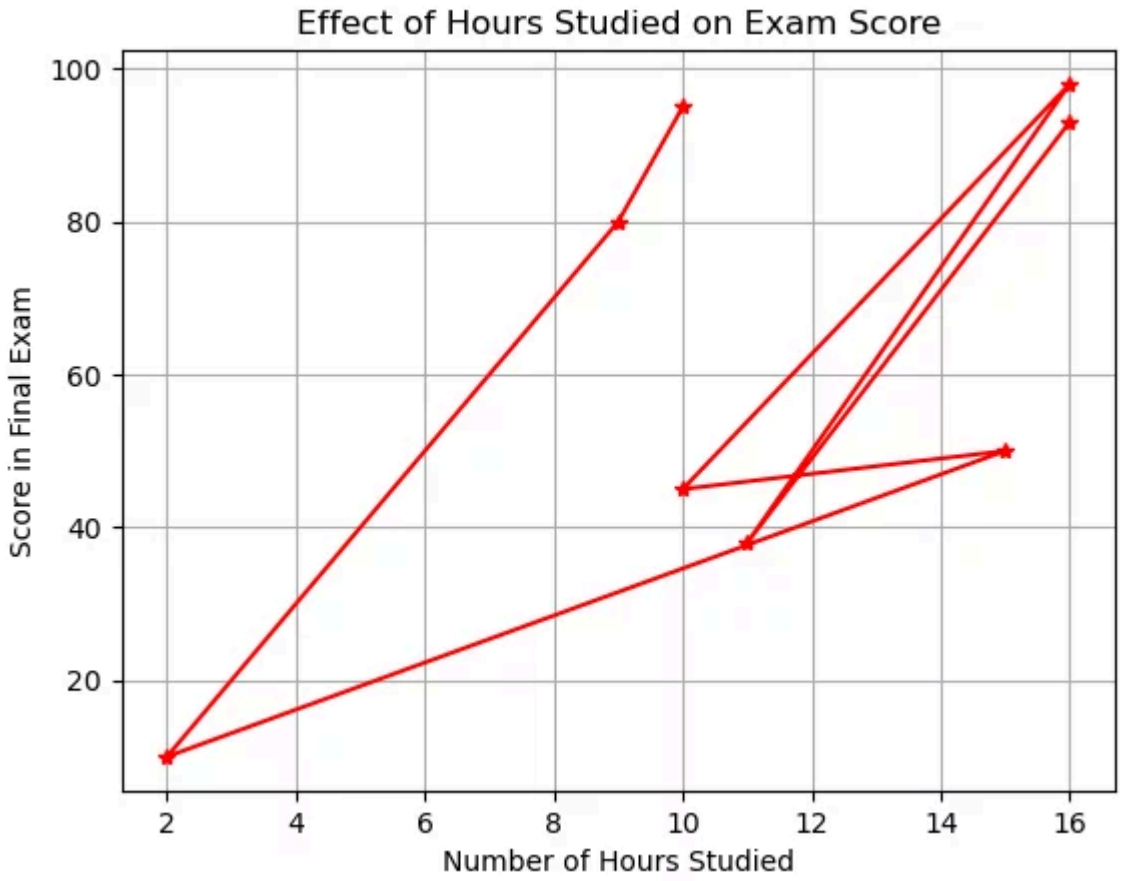
```
# Adding labels and title
plt.xlabel('Number of Hours Studied')
plt.ylabel('Score in Final Exam')
plt.title('Effect of Hours Studied on Exam Score')

# Displaying the plot
plt.grid(True)
plt.show()
```

To run this program online click on the link below and use Google Colab to run this program

M1P3 Program

Output



The program above demonstrates a clear trend: generally, the more hours students study, the better they perform on the final exam. However, there are some cases where this relationship isn't quite as straightforward, yielding slightly different outcomes.

Question 4

Histogram to check the frequency distribution

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)

Python Code

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
mtcars = pd.read_csv('mtcars.csv')  # Replace 'path_to_your_mtcars.csv' with the ac

# Plotting the histogram
plt.hist(mtcars['mpg'], bins=10, color='skyblue', edgecolor='black')

# Adding labels and title
plt.xlabel('Miles per gallon (mpg)')
plt.ylabel('Frequency')
plt.title('Histogram of Miles per gallon (mpg)')

# Displaying the plot
plt.show()
```

The required file `mtcars.csv` can be found below

mtcars.csv Download

To run this program online click on the link below and use Google Colab to run this program

M1P4 Program

Output

Module 2

Question 1

Kaggle Book Data set

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

Python Code

```
import pandas as pd
import numpy as np

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

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

# Find and drop the columns which are irrelevant for the book information
irrelevant_columns = ['Edition Statement', 'Corporate Author', 'Corporate Contribut
df.drop(columns=irrelevant_columns, inplace=True)

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

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

# Combine str methods with NumPy to clean columns
df['Place of Publication'] = np.where(df['Place of Publication'].str.contains('Lond

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

The required file `BL-Flickr-Images-Book.csv` can be found below

BL-Flickr-Images-Book.csv Download

To run this program online click on the link below and use Google Colab to run this program

M2P1 Program

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             Publisher  \
Identifier
206                      London               1879      S. Tinsley & Co.
216                      London               1868         Virtue & Co.
```

```
218                      London  1869  Bradbury, Evans & Co.
472                      London  1851           James Darling
480                      London  1857    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...
```

Module 3

Question 1

Logistic Regression

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.

Python Code

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline

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

```
y = iris.target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Create a pipeline with StandardScaler and LogisticRegression with regularization
pipeline = make_pipeline(StandardScaler(), LogisticRegression(C=1e4, max_iter=1000)

# Train the model
pipeline.fit(X_train, y_train)

# Calculate the accuracy on the testing set
accuracy = pipeline.score(X_test, y_test)
print("Classification accuracy:", accuracy)
```

To run this program online click on the link below and use Google Colab to run this program

M3P1 Program

Output

```
Classification accuracy: 1.0
```

Question 2

SVM classifier

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 RBF-kernel, 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.

Python Code

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
```

```
from sklearn.svm import SVC

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

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Set of hyperparameters to try
hyperparameters = [
    {'kernel': 'rbf', 'gamma': 0.5, 'C': 0.01},
    {'kernel': 'rbf', 'gamma': 0.5, 'C': 1},
    {'kernel': 'rbf', 'gamma': 0.5, 'C': 10}
]

best_accuracy = 0
best_model = None
best_support_vectors = None

# Train SVM models with different hyperparameters and find the best accuracy
for params in hyperparameters:
    model = SVC(kernel=params['kernel'], gamma=params['gamma'], C=params['C'], deci
    model.fit(X_train, y_train)
    accuracy = model.score(X_test, y_test)
    support_vectors = model.n_support_.sum()
    print(f"For hyperparameters: {params}, Accuracy: {accuracy}, Total Support Vect
    if accuracy > best_accuracy:
        best_accuracy = accuracy
        best_model = model
        best_support_vectors = support_vectors

print("\nBest accuracy:", best_accuracy)
print("Total support vectors on test data:", best_support_vectors)
```

To run this program online click on the link below and use Google Colab to run this program

M3P2 Program

Output

```
For hyperparameters: {'kernel': 'rbf', 'gamma': 0.5, 'C': 0.01}, Accuracy: 0.3, Tot
For hyperparameters: {'kernel': 'rbf', 'gamma': 0.5, 'C': 1}, Accuracy: 1.0, Total
For hyperparameters: {'kernel': 'rbf', 'gamma': 0.5, 'C': 10}, Accuracy: 1.0, Total
```

```
Best accuracy: 1.0
Total support vectors on test data: 39
```

Module 4

Question 1

Decision Tree based ID3 algorithm

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

Python Code

```
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd
from io import StringIO
from IPython.display import Image
import pydotplus

# Define the dataset
data = {
    'Price': ['Low', 'Low', 'Low', 'Low', 'Low', 'Med', 'Med', 'Med', 'Med', 'High'
    'Maintenance': ['Low', 'Med', 'Low', 'Med', 'High', 'Med', 'Med', 'High', 'High
    'Capacity': ['2', '4', '4', '4', '4', '4', '4', '2', '5', '4', '2', '2', '5'],
    'Airbag': ['No', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'Yes', 'Yes
```

```
    'Profitable': [1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1]
}

df = pd.DataFrame(data)

# Convert categorical variables into numerical ones
df = pd.get_dummies(df, columns=['Price', 'Maintenance', 'Airbag'])

# Separate features and target variable
X = df.drop('Profitable', axis=1)
y = df['Profitable']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Create a decision tree classifier
clf = DecisionTreeClassifier(criterion='entropy')

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Predict on the testing data
y_pred = clf.predict(X_test)

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

# Visualize the decision tree
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data, filled=True, rounded=True, special_characte
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

To run this program online click on the link below and use Google Colab to run this program

M4P1 Program

Output

```
Accuracy: 0.6666666666666666
```

Question 2

Clustering

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.

Python Code

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

# Load the dataset
data = np.loadtxt("Spiral.txt", delimiter=",", skiprows=1)
X = data[:, :2]  # Features
y_true = data[:, 2]  # Actual cluster labels

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

# K-means clustering
# kmeans = KMeans(n_clusters=3, random_state=42)
kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
kmeans_clusters = kmeans.fit_predict(X)

# Single-link Hierarchical Clustering
single_link = AgglomerativeClustering(n_clusters=3, linkage='single')
single_link_clusters = single_link.fit_predict(X)

# Complete-link Hierarchical Clustering
complete_link = AgglomerativeClustering(n_clusters=3, linkage='complete')
complete_link_clusters = complete_link.fit_predict(X)

# Compute the Rand Index
rand_index_kmeans = adjusted_rand_score(y_true, kmeans_clusters)
rand_index_single_link = adjusted_rand_score(y_true, single_link_clusters)
```

```
rand_index_complete_link = adjusted_rand_score(y_true, complete_link_clusters)

print("Rand Index for K-means Clustering:", rand_index_kmeans)
print("Rand Index for Single-link Hierarchical Clustering:", rand_index_single_link
print("Rand Index for Complete-link Hierarchical Clustering:", rand_index_complete_

# This code will compute the Rand Index for each clustering method and provide a vi
# The Rand Index ranges from 0 to 1, where 1 indicates perfect clustering agreement
# The method with a higher Rand Index is better at recovering the true clusters.
```

To run this program online click on the link below and use Google Colab to run this program

M4P2 Program

Output

Module 5

Mini Project

Simple web scrapping in social media

Python Code

```
import requests
from bs4 import BeautifulSoup

# URL of the Instagram profile you want to scrape
url = 'https://www.instagram.com/openai/'

# Send a GET request to the URL
response = requests.get(url)

print(response.status_code)

# Check if the request was successful (status code 200)
if response.status_code == 200:
    # Parse the HTML content of the page
    soup = BeautifulSoup(response.text, 'html.parser')

    # Find all post elements
    posts = soup.find_all('div', class_='v1Nh3')

    # Extract data from each post
    for post in posts:
        print("Hi")
        # Extract post link
        post_link = post.find('a')['href']

        # Extract post image URL
        image_url = post.find('img')['src']

        print(f"Post Link: {post_link}")
        print(f"Image URL: {image_url}")
        print("------")
else:
    print("Failed to retrieve data from Instagram")
```

To run this program online click on the link below and use Google Colab to run this program

M5P1 Program

Output

[Setting up Anaconda Python Programming Environment on Ubuntu](#)

February 24, 2023

Programming

[VTU Lab Manuals using FOSS](#)

October 18, 2023

Programming

[Data Visualization with Python – BCS358D](#)

October 13, 2023

Programming

Prabodh C P

Prabodh C P is a faculty in the Dept of CSE SIT, Tumkur and also currently a Research Scholar pursuing PhD in IIT Hyderabad. He conducts online classes for C, C++, Python. For more info call +919392302100

Categories: Programming

Tags: 21AD62, 6th semester, Anaconda Python Distribution, CSE, Data Science, DATA SCIENCE AND ITS APPLICATIONS, FOSS, Histogram, Jupyter Notebook, Lab Manual, Linear Plot, Matplotlib, Numpy, Pandas, Python, Python Program, Solution Manual, Spyder, VI semester, VTU

Leave a Comment