AI Lab Sheet
III Year / II Part
Faculty: Computer

Labsheet#7: Machine Learning using Google Colab

Objective

1. Familiarization with Google Colab, Python, numpy, pandas, matplotlib, tensorflow, keras & Machine Learning.

Google Colab

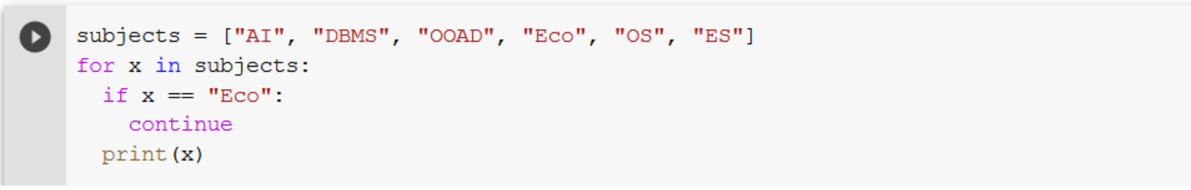
Learn - How to use Google Colab?

- https://www.analyticsvidhya.com/blog/2020/03/google-colab-machine-learning-deep-learning/
- https://www.tutorialspoint.com/google_colab/index.htm

Start Your Notebook on https://colab.research.google.com

Experiment#1: Run the following codes in Google Colab and show output by modification these programs to atomic form.

E1.1 Loops & continue in Python



```
subjects = ["AI", "DBMS", "OOAD", "Eco", "OS", "ES"]
for x in subjects:
  if x == "Eco":
    continue
  print(x)
```

E1.2 Matrix Multiplication using Nested Loop

```
X = [[1,2,3],
    [4 ,5,6],
    [7 ,8,9]]
Y = [[2,4,6,8],
    [1,3,5,7],
    [4,5,6,8]]
# result is 3x4
result = [[0,0,0,0],
         [0,0,0,0],
         [0,0,0,0]]
for i in range(len(X)):
   for j in range(len(Y[0])):
       for k in range(len(Y)):
           result[i][j] += X[i][k] * Y[k][j]

for r in result:
   print(r)
```

E1.3 Write a function to display prime no as per user’s wish.

```
def generatePrimes(n):
  # Initialize a list
  primes = []
  for possiblePrime in range(2, n + 1):
  # Assume number is prime until shown it is not.
    isPrime = True
    for num in range(2, int(possiblePrime ** 0.5) + 1):
      if possiblePrime % num == 0:
        isPrime = False
        break

      if isPrime:
        primes.append(possiblePrime)

  print(primes)

n = int(input("Enter n to display primes 1 to n:"))
generatePrimes(n)
```

Assignment 1.1: Modify the program E1.3 to atomic level.

Experiment#2: Copy & run the following codes in Google Colab:

E2.1 3D Plot

```
from mpl_toolkits import mplot3d
import numpy as np
import matplotlib.pyplot as plt
x = np.outer(np.linspace(-2, 2, 30), np.ones(30))
y = x.copy().T # transpose
z = np.cos(x ** 2 + y ** 2)

fig = plt.figure()
ax = plt.axes(projection='3d')

ax.plot_surface(x, y, z,cmap='viridis', edgecolor='none')
ax.set_title('Surface plot')
plt.show()
```

E2.2 Correllogram

```
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

# Import Dataset
df = pd.read_csv("https://raw.githubusercontent.com/ErSKS/AI/master/Datasets/mtcars.csv")

# Plot
plt.figure(figsize=(12,10), dpi= 80)
sns.heatmap(df.corr(), xticklabels=df.corr().columns, yticklabels=df.corr().columns, cmap='RdYlGn', center=0, annot=True)

# Decorations
plt.title('Correlogram of mtcars', fontsize=22)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```

E2.3 Pairwise Plot

```
import matplotlib.pyplot as plt
import seaborn as sns
# Load Dataset
df = sns.load_dataset('iris')

# Plot
plt.figure(figsize=(10,8), dpi= 80)
sns.pairplot(df, kind="scatter", hue="species", plot_kws=dict(s=80, edgecolor="white", linewidth=2.5))
plt.show()
```

Assignment 2.1: Perform following task for the dataset https://github.com/ErSKS/AI/blob/master/Datasets/mppl_2021.csv

- Display Top Rows
- Draw a Pie Chart
- Draw a Bar Graph in Orange Color

Assignment 2.2: What type of plots can be used for comparison, distribution, composition & relationship? Demonstrate your code with respective plot.

Experiment#3: Deep Learning Model Life-Cycle

5-Step Model Life-Cycle

The five steps in the life-cycle are as follows:

1. Define the model.
2. Compile the model.
3. Fit the model.
4. Evaluate the model.
5. Make predictions.

E3.1 Training a neural network on MNIST with Keras

Explore:

- https://github.com/ErSKS/Colab/blob/main/05_Convolution_Keras.ipynb

Reference:

- https://colab.research.google.com/github/tensorflow/datasets/blob/master/docs/keras_example.ipynb
- https://machinelearningmastery.com/tensorflow-tutorial-deep-learning-with-tf-keras/

Assignment 3.1: Develop the multi-class classifier model using any beautiful algorithm. And explain why you choose the algorithm.

Machine Learning Notebooks

Explore: https://github.com/ageron/handson-ml2

Fun with Image Classifier

Explore: https://teachablemachine.withgoogle.com/train/image