**Github :**

****

**DAY1 :12-02-2024**

**IDE:Integrated development Environment**

**Google Colab->**

**numpy-python library for numerical computing it supports arrays,matrices,scientific computing,data analysis,machine learning.**

**Applications of numpy:**

**1)data analysis**

**2)machine learning.**

**Functions in numpy:**

**1)array**

**2)zeros**

**3)ones**

**4)arange**

**5)reshape**

**6)slice[a:b]**

**7)add**

**vstack->vertically add elements**

**stack->horizontally add elements**

**array.split(array1,index)**

**array.transpose(matrix)**

**Linear algebra with numpy:**

**1)dot product**

**2)linalg->eigen values eigenand vectors**

**axis=0->row;**

**axis=1->column**

**sum->sum of all the elements in the matrix**

**Statistical algebra:**

**1)mean**

**2)median**

**3)standard deviation**

**4)Variance**

**Data:**

**(load txt**

**savetxt)->data access in colab**

**matplotlib->**

**1)plot**

**2)confusion matrix**

**3)data pre processing ->it is used for the mismatching the data.**

**DAY2 :13-02-2024**

**Pandas**-Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

**#importing pandas**

import pandas as pd

a=['Mahesh','farhan','Akhil','Sai','Kamal','Gafoor','lova']

r=pd.Series(a,index=[39,36,35,16,8,57,44])

print(r)

**#read csv file and text file**

import pandas as pd

import numpy as np

data=pd.read\_csv("/tableConvert.com\_gyhh6h.csv") #csv file

print(data)

**#printing one row as series**

print(data.loc[1])

**#read XLSX file**

import pandas as pd

data=pd.read\_excel("/content/salesworkload.xlsx",sheet\_name=1)

print(data)

**#drive.mount():**

import pandas as pd

from google.colab import drive

drive.mount('/content/drive')

**#Taking 10 values for manual testing**

data\_testing=data.tail(10)

for i in range(21416,21416,-1):

data.drop([i],axis=0,implace=True)

**#Concatination**:obtaining a new string that contains both of the original strings(concat())

import pandas as pd

import numpy as np

data=pd.read\_csv("/content/tableConvert.com\_gyhh6h.csv")

data.shape #shape():obtain the shape of a DataFrame(rows,columns)

data\_manual=data.tail(9)

for i in range(1,0,-1):

data.drop([i],axis=0,inplace=True)

data\_manual=pd.concat([data],axis=0)

data\_manual.to\_csv("Last\_10\_lines\_csv")

**#groupby()**:it is used to group data based on the same value in a specific column

print(data\_manual.groupby(['sum'])['Average'plt.show()

**#Plotting of graph using "matplotlib.pyplot"**

import numpy as np

import matplotlib.pyplot as plt

runs=np.array([100,50,91,78,89,25,34,19,9,10])

w=np.array([1,0,2,0,3,7,8,9,7,5])

plt.scatter(runs,w,color='blue') #scatter() is used to create a scatter plot by using dots

plt.title('IndvsAus\_score') #title():its gives the title for plot

plt.show() #show():it displays the plot on screen

**#Generate array of 200 values between -pi & pi**

import numpy as np

import matplotlib.pyplot as plt

tiger=np.linspace(-2\*np.pi,2\*np.pi,100) #linspace():create an evenly spaced sequencd

print(tiger)

plt.plot(tiger,np.sin(tiger),color='black') #SIN plt.plot(x,y,,color,label....)

plt.title("sin(x)")

**#Display plot**

plt.show()

**#combining two graphs**

import numpy as np

import matplotlib.pyplot as plt

**#creating x**

overs=np.arange(5,50,5)

overs\_a=np.arange(5,30,5)

**#creating y**

runs\_i=np.array([25,51,84,131,160,189,220,250,267])

runs\_a=np.array([15,41,94,100,151])

wickets=np.array([12,32,96])

**#plotting**

plt.plot(overs,runs\_i,color='blue',label='India')

plt.plot(overs\_a,runs\_a,color='yellow',label='Aus')

plt.legend(loc='best') #combining two graphs

**#displaying the final graph**

plt.show()

**#subplot()**:create a grid of subplots within a single figure

import numpy as np

import matplotlib.pyplot as plt

a=[230,560,780,127,128]

b=[200,160,270,127,400]

years=[1,2,3,4]

profit\_a=[(a[i]-a[i-1]) for i in range(1,len(a))]

profit\_b=[(b[i]-b[i-1]) for i in range(1,len(b))]

plt.subplot(2,1,2)

plt.plot(years,profit\_a,color='hotpink',linewidth='3',label='companyA',marker='+',ms='15',mec='k')

plt.legend(loc='best')

plt.subplot(2,1,1)

plt.plot(years,profit\_b,color='black',linestyle='dotted',label='companyB',marker='\*')

plt.legend(loc='best')

plt.show()

**#pie chart**

import numpy as np

import matplotlib.pyplot as plta=np.array([25,60,5,10])

labe=["AIML","PYTHON","PANDAS","NUMPY"]

explo=[0.2,0,0,0]

plt.pie(a, labels=labe, explode=explo, startangle=180,textprops={'fontsize':44})

plt.show()

\***Startangle**:rotates the plot by the specified degrees in counter clockwise direction

\***explode**:part of pie chart will be separated from the main pie.

**Day 3: 14-02-2024**

**Seaborn:**python library for making statistical graphics in Python.

**#installing seaborn library**

pip install seaborn

**#create a scattter plot**

import seaborn as sns

import matplotlib.pyplot as plt

**#Load example dataset**

tips=sns.load\_dataset("tips")

sns.scatterplot(x="total\_bill",y="tip",data=tips)

sns.violinplot(x="total\_bill",y="tip",data=tips)

plt.title("Scatter Plot of Total Bill vs. Tip")

plt.xlabel("Total Bill($)")

plt.ylabel("Tip ($)")

plt.show()

**#create a scattter plot using csv file**

import seaborn as sns

import pandas as pd

import matplotlib.pyplot as plt

tips=pd.read\_csv("/content/city\_temperature(0).csv")

print(tips)

sns.scatterplot(x="Year",y="AvgTemperature",data=tips)

plt.title("Scatter Plot of Year vs. Temperature")

plt.xlabel("Year")

plt.ylabel("Temperature (c)")

plt.show()

**#compute correlation matrix**

import seaborn as sns

import matplotlib.pyplot as plt

**#Load example dataset**

iris=sns.load\_dataset("iris")

print(iris)

**#compute correlation matrix**

correlation\_matrix=iris.corr()

**#create a heatmap of the correlation matrix**

sns.heatmap(correlation\_matrix,annot=True,cmap="coolwarm")

plt.title("Correlation heatmap of Iris Dataset")

print(iris)

**Machine learning** **:-**

**Machine learning can be done in 3 steps**

**1)training**

**2)testing**

**3)processing**

**Types of machine learning**

**1)supervised machine learning : labelled**

**2)unsupervised machine learning : unlabelled**

**3)semi-supervised machine learning : both labelled and unlabelled**

**NEURAL NETWORK:** **inter-connection of the neurons**

**CNN: A convolutional neural network(CNN) is a type of deep learning.neural network architecture commonly used in computer vision.computer vision is a field of artificial intelligence that enables a computer to understand and interpret the image or visual data .**

**Types of layers:**

**1.input layer**

**2.hidden layer**

**convolutional layer**

**Activation layer -max pooling layer**

**average layer**

**dense layer**

**3.output layer**

**1.INPUT LAYER: Its the layer in which we give input to our model.In CNN,generally,the input will be an image or a sequence of images**

**.**

**2.CONVOLUTIONAL LAYER :This is the layer,which is used to extract the feature from input dataset.it applies a set of learnable filters known as the kernels/filters to the input images.**

**the output of this layer is referred ad feature maps.suppose we use a total of 12 filters for this layer we’ll get an output volume of dimension 32x32x12**

**3.ACTIVATION LAYER:- By adding an activation function to the output of the preceding layer,activation layers add nonlinearity to the network.it will apply an element wise activation function to the output of the convolution.**

**ACTIVATION FUNCTION: The activation function decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it.the purpose of the activation function is to introduce non-linearity into the output of a neuron.**

**Activation function make the back-propagation possible since the gradients are supplied along with the error to update the weights and biases.**

**TYPES OF ACTIVATION FUNCTION:**

**1)tanh** - **range: -1 to +1,nature-nonlinear,hidden layer,back-propagation**

**2)sigmoid** - **A=1/(1+e-x) formula,range=0 to 1,it is used in output layer of a binary classification.0.5 less then and 0.5 greater than**

**3)relu** - **formula A(x)=max(0,x),range : [0,infinity],nature:- non-linear, multiple layers of neuron being activated by the Relu function.it gives fast response and calculations(computation).it is the best for error corrected fastly. It is a rectified linear,hidden layer**

**4)softmax** - **nature : non-linear,output layer,it can handle multi-class classification problems,range: 0 to 1.it is very useful to predict the probability.**

**4.POOLING LAYER**

**This layer is periodically inserted in the convnets and its main function is to reduce the size of volume which makes the computation fast, reduce memory and also prevents overfitting.**

**POOLING LAYERS are two types:**

**1.max pooling layer-16x16x12**

**2.average pooling layer**

**5.OUTPUT LAYER:**

**The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.**

**Day 4:15-02-2024**

**Find Mean:-**

**Find the mean of both dependent and independent Variable:**

**X’ = 5+7+12+16+20/5(ind)**

**Y’ = 40+120+180+210+240/5(dep)**

**Find differences between each x point and x-mean (x-x’)**

**Find differences between each y point and y-mean (y-y’)**

**Find sum of squares of (x-x’) Sum(x-x’)**

**Find product of (x-x’)and**

**Linear Regression:**

**>Linear regression is also a type of machine learning algorithm**

**>Supervised machine-learning algorithm.**

**>Learns from the labelled datasets and maps the data points to the most optimized linear functions.**

**>These points can be used for prediction on new datasets.**