**Documentation**

**DAY 1- 23-02-2024**

**MODULE-1 :**

**IDE - INTEGRATED DEVELOPMENT ENVIRONMENT**

**GOOGLE COLAB**

**1)Numpy-python library it is used for numerical computing.it supports arrays,matrices,scientific computing data analysis,machine learning**

**Major applications of numpy:**

**1)Data Analysis.**

**2)Scientific computing**

**Array Manipulation:**

**1)Reshaping of an Array**

**2)Slicing the Array**

**3)Stacking the array in vertical using “vstack” and horizontal using “Stack”**

**4)Splitting of an array using “split”**

**Mathematical operations on Array:**

**1)Addition of a Arrays**

**2)Subtraction of Arrays**

**3)Multiplication of Arrays**

**4)Transpose of a matrix**

**Linear Algebra with Numpy:**

**1)Matrix Multiplication**

**2)Calculation of Eigen value and eigen vector**

**3)Sum of elements in a matrix**

**row wise using ‘axis=0’ and colum ‘axis=1’**

**Statistical Operations**

**1)mean**

**2)median**

**3)variance**

**4)standard variance**

**Data:**

**Loadtxt**

**savetxt**

**Libraries:**

**Matplotlib:1.plot 2.confusion matrix 3.data pre processing(used for mismatching data)**

**Seaborn**

**Binary classification-it has only 2 classes either 0 or 1(T or F)**

**Multi classification- +ve(1),-ve(0),neutral(0.5)**

**Confusion matrix-11,00,10,01(completely true,completely false,not completely true or false,not completely false or true)**

**DAY-2 : 24-02-2024**

**2)Pandas-data manipulation package in python for tabular data.that is data in the form of rows and columns also know as Dataframes.**

**.Series() is used to allocate index number.**

**.read\_csv()-to read the csv file**

**.read\_excel()-to read the xlsr**

**.loc()-to print the specific location**

**.fillna()-any row or any column missing value gets filled with 0.**

**.drop\_duplicates()-deletes the duplicates.**

**.head()-displays from top up to given number of rows,if not specified takes starting 5 rows.**

**.tail()-displays from bottom for given number of rows,if not specified takes last 5 rows.**

**#from google.colab import drive#**

**#drive.mount('/content/drive')#**

**Linking drive to the colab we use mounting**

**.shape()-displays the rows and columns of the given file.**

**.concat()-to concatenate the two files for testing.**

**.groupby()-it displays the given column it groups the given data and display once.**

**.count()-it displays the count of duplicate values which are grouped in the column.**

**DAY:- 3 (26-02-2024)**

**MATPLOTLIB :-**

**Most of the Matplotlib utilities lies under the pyplot submodule, and are usually imported under the plt**

**Import matplotlib.pyplot as plt**

**Ex:-**

**Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**plt.plot(x,y)**

**plt.show()**

**1. The scatter() function plots one dot for each observation. It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis.**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**plt.scatter(x,y)**

**plt.show()**

**2. color or the c argument**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**plt.scatter(x,y,color=”yellow”)**

**plt.show()**

**3. size argument :-**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**s=np.array[1,2,3,4,5]**

**plt.scatter(x,y,size=s)**

**plt.show()**

**4. marker to emphasise each point with a specified marker**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**plt.scatter(x,y,marker=’o’)**

**plt.show()**

**5. shortcut string notation parameter to specify the marker.This parameter is also called**

**fmt, and is written with this syntax:**

**Marker|line|color**

**Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**plt.scatter(x,y,o:r) // marker=o,line=dotted,color=red**

**plt.scatter(x,y,o-r) // line=solid line**

**plt.scatter(x,y,o--r) // line=dashed line**

**plt.scatter(x,y,o-.r) // line=dashed,dotted**

**plt.show()**

**6. Marker size(ms):-**

**Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**plt.scatter(x,y,marker=’o’,ms=5)**

**plt.show()**

**7.markeredgecolor (mec):-**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**plt.scatter(x,y,marker=’o’,ms=5,mec=’r’)**

**plt.show()**

**8.markerfacecolor (mfc):-**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array[1,2,3,4,5]**

**y=np.array[1,2,3,4,5]**

**plt.scatter(x,y,marker=’o’,ms=5,mfc=’r’)**

**plt.show()**

**9) line arguments:-**

**1) linestyle:- plt.scatter(x,y,linestyle=’dotted’)**

**2) linecolor:- plt.scatter(x,y,color=”yellow”)**

**3) linewidth:- plt.scatter(x,y,linewidth=5)**

**10.xlabel() :- give name to x axis Ex:- plt.xlabel(“x label name”)**

**11)ylabel() :- give name to y axis Ex:- plt.ylabel(“y label name”)**

**12.title() :- give the title for the graph Ex:- plt.title(“title name”)**

**13.pie():-function to draw pie charts**

**Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array([1,2,3,4,5])**

**plt.pie(x)**

**plt.show()**

**14.label parameter:-must be an array with one label for each wedge**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array([1,2,3,4,5])**

**y=[“1”,”2”,3”,”4”,”5”]**

**plt.pie(x,labels=y)**

**plt.show()**

**15.The explode parameter, if specified, and not none, must be an array with one value for each wedge**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array([1,2,3,4,5])**

**y=[“1”,”2”,3”,”4”,”5”]**

**z=[0.2,0,0,0,0]**

**plt.pie(x,labels=y,explode=z)**

**plt.show()**

**16. shadow parameter:-**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array([1,2,3,4,5])**

**y=[“1”,”2”,3”,”4”,”5”]**

**z=[0.2,0,0,0,0]**

**plt.pie(x,labels=y,explode=z,shadow=True)**

**plt.show()**

**17. color parameter:-**

**Ex:- Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array([1,2,3,4,5])**

**y=[“1”,”2”,3”,”4”,”5”]**

**p=[“black”,”yellow”,”red”,”blue”,”green”]**

**z=[0.2,0,0,0,0]**

**plt.pie(x,labels=y,colors=p)**

**plt.show()**

**14.bar():-function to draw bar graphs**

**Import matplotlib.pyplot as plt**

**Import numpy as np**

**x=np.array([1,2,3,4,5])**

**plt.bar(x)**

**plt.show()**

**SEABORN:- This library based on matplotlib**

**1.by importing seaborn as sns**

**Ex:- Import seaborn as sns**

**2. we load the data set by using**

**Import seaborn as sns**

**sns.load\_dataset(“dataset name”)**

**3. different methods:-**

**1)barplot():- bar graph**

**Ex:- Import matplotlib.pyplot as plt**

**Import seaborn as sns**

**sns.load\_dataset(“tips”)**

**sns.barplot(x=”days”,y=”total\_bill”,data=”iris”)**

**plt.show()**

**2)boxplot():-**

**Ex:- Import matplotlib.pyplot as plt**

**Import seaborn as sns**

**sns.load\_dataset(“tips”)**

**sns.boxplot(x=”days”,y=”total\_bill”,data=”iris”)**

**plt.show()**

**3)violin plot():-**

**Ex:- Import matplotlib.pyplot as plt**

**Import seaborn as sns**

**sns.load\_dataset(“tips”)**

**sns.violinplot(x=”days”,y=”total\_bill”,data=”iris”)**

**plt.show()**

**4)lineplot():-**

**Ex:- Import matplotlib.pyplot as plt**

**Import seaborn as sns**

**sns.load\_dataset(“tips”)**

**sns.lineplot(x=”days”,y=”total\_bill”,data=”iris”)**

**plt.show()**

**5)heatmap():-**

**Ex:- Import matplotlib.pyplot as plt**

**Import seaborn as sns**

**sns.load\_dataset(“tips”)**

**sns.heatmap(x=”days”,y=”total\_bill”,data=”iris”)**

**plt.show()**

**6)jointplot():-**

**Ex:- Import matplotlib.pyplot as plt**

**Import seaborn as sns**

**sns.load\_dataset(“tips”)**

**sns.jointplot(x=”days”,y=”total\_bill”,data=”iris”)**

**plt.show()**

**6)countplot():-**

**Ex:- Import matplotlib.pyplot as plt**

**Import seaborn as sns**

**sns.load\_dataset(“tips”)**

**sns.countplot(x=”days”,y=”total\_bill”,data=”iris”)**

**plt.show()**

**7)lm plot():-**

**Ex:- Import matplotlib.pyplot as plt**

**Import seaborn as sns**

**sns.load\_dataset(“tips”)**

**sns.lmplot(x=”days”,y=”total\_bill”,data=”iris”)**

**plt.show()**

**DAY-4 (27/02/2024)**

**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.If it is less than 0.5 then it is considered as 0 else 1.**

**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- 6 (28-02-2024)**

**Linear regression**

**Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features.**

**Simple linear regression formula**

**The formula for a simple linear regression is:y=a+bx**

**y is the predicted value of the dependent variable (y) for any given value of the independent variable (x).**

**a is the intercept, the predicted value of y when the x is 0.**

**b is the regression coefficient – how much we expect y to change as x increases.**

**x is the independent variable ( the variable we expect is influencing y).**

**Logistic regression :-**

**Logistic regression is used for binary classification where we use a sigmoid function that takes input as independent variables and produces a probability value between 0 and 1.**

**For example, we have two classes Class 0 and Class 1 if the value of the logistic function for an input is greater than 0.5 (threshold value) then it belongs to Class 1 it belongs to Class 0. It’s referred to as regression because it is the extension of linear regression but is mainly used for classification problems**