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

**-IDE**-Integrated development environment(Google colab)

Github repository creation(folder name=DATA ANALYSIS)

Review of core python concepts: data types,operators,control flow,functions,modules,package

* Data Types:
  1. Python has several built-in data types:
     1. **Numeric Types**: Integers (int), floating-point numbers (float), and complex numbers (complex).
     2. **Text Type**: Strings (str).
     3. **Sequence Types**: Lists (list), tuples (tuple), and ranges (range).
     4. **Mapping Type**: Dictionaries (dict).
     5. **Set Types**: Sets (set) and frozen sets (frozenset).
     6. **Boolean Type**: Booleans (bool).
     7. **Binary Types**: Bytes (bytes), byte arrays (bytearray), and memory views (memoryview).
     8. **None Type**: Represents the absence of a value (None).

# Operators:

* 1. Python supports various operators, including arithmetic, comparison, logical, and assignment operators.

# Control Flow:

* 1. Python provides control flow statements like if, for, and while.
  2. You can use break, continue, and else clauses with loops.
  3. Functions allow you to define reusable code blocks.

# Modules and Packages:

* 1. Modules are files containing Python code that can be imported into other programs.
  2. Packages are directories containing multiple modules.

NUMPY

* Numpy is used for numerical or scientific computation in arrays,vectors and matrices.
* To import the numpy package we just have to use the import statement which is as follows: import numpy as np(our wish).

# Functions in numpy

## ARRAY CREATION:

* **np.array()**- creates an array from a list
* **np.zeros()**- creates an array filled with zeros of the specified size
* **np.ones()** - creates an array filled with ones of the specified shape.
* **np.arange()**-Generates numbers up to the given value and we can also starting value,ending value and step value
* For all these functions the default data type is “float” and we can change this using “dtype”.
* ARRAY MANIPULATION:
  + **reshape()**-Reshapes the array into desired shaped array
  + slicing-Slicing is done with **“:”** operator.It prints the elements from the given range.
    - Ex:- b=a.reshape(r,c)

a = created array

r = number of rows

c = number of columns

* + **Transpose**-Transpose is done as follows:**”array\_variable.T”** As the name it returns the transpose.
* Ex:- b=a.T

**->** a = created array

* + **np.split(a,n)**-Split function splits the given array(a) into n arrays

1. Ex:- b=np.split(a,n)

**->** a = created array

**->** n = number of divisions/split

* + **np.dot(a,b)-**Gives the dot product of matrix a and b

Ex:- c=np.dot(a,b)

**->** a=first matrix stored variable

**->** b=second matrix stored variable

* + **np.linalg.eig(a)-**Gives the eigenvalues and vectors of given matrix

1. Ex:- b=np.linalg.eig(a)

**->** a = created array

* + **np.loadtxt(path,dtype)-**Loads any file from the path that we’ve given and we can give the datatype want like**”dtype=int”**
  + **np.savetxt(path,file)-**Loads the file that we’ve given into the path
  + **np.random.rand()-**Randomly produces numbers between 0 and 1
  + **np.random.randint(a,b)-**Randomly produces between a and b
  + **type(a)-**Gives the type of the matrix a
  + ndim and shape are used to find the dimension of array and shape the array respectively
    - Invoked as **“Array\_variable.ndim”** and **“Array\_var.shape”**
  + Element multiplication is done using **“\*”**operator and Matrix multiplication is done using “@”.
    - EX:a\*b and a@b
  + **sum()-**Returns the sum of all the elements in the matrix
    - EX:a.sum() where a is the matrix
  + **max()-**Returns the maximum element in the matrix
  + **cumsum()-**Returns the cumulative sum of the elements
  + The above three functions have special feature called **“axis”** if axis=1 respective operation is done along the column and if axis=0 then the operations are done along the rows
  + **np.vstack(a**,**b) and np.hstack(a.b)-**Used to stack the given two matrices a and b vertically and horizontally respectively.
  + **np.dstack(a)-**It used for some changes in matrix a.The changes are:
    - Number of rows become number of groups
    - Number of column become number of rows
    - Number of groups become number of columns
* **np.eye(n)-**It gives the identity matrix of size **‘n’.**
* **np.full(size,n)-**It produces a matrix of given **“size”** with the element **“n”.**
* **np.asarray(x)-**It turns the list **“x”** into a symmetric array or matrix.
* **np.inner(x,y)-**It produces the inner product of two vectors or arrays **x** and **y**
* **np.outer(x,y)-**It produces the outer product of two vectors or arrays **x** and **y**
* **np.cross(x,y)-**It produces the cross product of two vectors or arrays **x** and **y**
* **np.rint(a)-**It gives the round value of given “**a**”.This **a** can be a value or an array.
* **np.true\_divide(a,b)-**It gives the exact quotient for the division betweenIt gives the exact quotient for the division between 1st element in matrix **a** and 1st element in matrix **b**.
* **np.unique(a)-**It removes all the repeated values from **“a”** and produces a unique list
* **np.union1d(a,b)-**It gives the union of two matrices or arrays **a** and **b**.
* **np.intersect1d(a,b)-**It gives the common elements(intersection) of two matrices or arrays **a** and **b**.
* **np.setdiff1d(a,b)-**It gives the set difference for the matrix or array **a** and **b**.
* **np.hypot(x,y)-**It calculates the hypothesis using the edges **x** and **y** that we give.
* **np.divmod(x,y)-**It divides the 1st element in matrix **x** and 1st element in matrix **y** and produces both the quotient and remainder.
* **np.mod(x,y)--**It divides the 1st element in matrix **x** and 1st element in matrix **y** and produces only the remainder.
* **np.divide(x,y)--**It divides the 1st element in matrix **x** and 1st element in matrix **y** and produces an exact quotient.
* **np.multiply(x,y)--**It multiplies the 1st element in matrix **x** and 1st element in matrix **y** and produces the result.

# Random functions:

* **random.normal([loc],[scale],size):**It produces a matrix of given “size” with random numbers.
  + **loc:-**mean
  + **Scale:-**standard deviation.These two are optional
* **random.binomial(n,p,size):n** is number of trail and **p** is the probability so it produces a series of size **“size”** based on **n** and **p**
* **random.poisson(lam,size):**It produces a series of size **“size**” based on poisson distribution.
* **random.choice(val,[p],[size]):**It chooses a value and given values(val) and we can also specify probability(**p**) and size of matrix(**size**) but these are optional.

# 

**DAY-2(24/02/2024)**

* Pandas:
* Pandas is a Python library essential for data manipulation and analysis.
* It simplifies tasks like reading various data formats, cleaning messy datasets, and performing complex data operations.
* It is rich in a set of statistical functions.
* Pandas also plays a vital role in data preprocessing for machine learning.
* We can import pandas into our code by using the import statement which is **import pandas as pd**(our wish).

Functions in pandas:

* **pd.series(x,[index]):**It produces **"x"** as a series with provided index values.
* **pd.read\_csv(path):**It loads a csv file from the path that we gave.
* **pd.read\_excel(path,sheet\_name):**It loads a xlsx file from the path that we gave.Sheet name is optional we can give the sheet number that we want to load.
* **pd.head(n)-**By default Head gives the first 5 rows but we can choose a specific number that we want(n).
* **pd.tail(n)-**By default tail gives the last 5 rows but we can choose a specific number that we want(n).
* **pd.describe()-**It gives both the first 5 and last 5 rows as output.
  + FOR THE BELOW df IS WHERE THE FILE LOADED
* **df.describe().T-**It also gives both first 5 and last 5 rows as output but it interchanges the rows and columns(Transpose).
* **df.shape[]-**By default it gives the number of rows and columns that the file contains but if we give **‘0’** as parameter then it gives only the row count and if we give **‘1’** it gives the count of columns
* **df.columns-**It gives the names of the columns.
* **df.copy()-**It copies the df file and stores it into some other.
  + EX:- df2=df.copy() Here df is copied in to df2
* **df.loc[ ]-**”.loc[]” is primarily label based.
  + Allowed inputs are:
    - A single label, e.g. 5 or 'a', (note that 5 is interpreted as a label of the index, and never as an integer position along the index).
    - A list or array of labels, e.g. ['a', 'b', 'c'].
    - A slice object with labels, e.g. 'a':'f'.
* **df.isnull()-**It returns boolean value if the given column or row is nan then it shows true or else false.
  + We can also use this with the **.sum()** function.It gives the number of nan values each column has.
    - The syntax is **df.isnull().sum()**
* **df.index==val-**It gives the row that we’ve given.
  + - **val**=row number
    - We can also with the help giving range that is as follows
      * df.index.isin(range(n1,n2)) then it prints rows from n1 to n2.
* **df.dropna()-**It is used to drop the row that contains none value.
  + But we can drop both rows and columns by using the below function.
    - df3.dropna(inplace=True,how='all')
* **df[‘column’].mean()-**It is used to find the mean for a specified column.
* **df.fillna(val)-**It is used to fill the none value with whatever the value that we specified.
  + **val**=The value that we want to fill.
* **df.drop\_duplicates()-**It drops repeated column
* **df.rename(columns = {'old name':'new name'})-**It renames the old column name with the given new column name.
  + We can do this in other way which is
    - df.columns=[new column name]
  + We can also create new columns by helping other columns.
* **df.groupby(column)-**When you apply the .groupby() function to a column,it returns a GroupBy object. This object allows you to perform various operations on the grouped data such as mean,max and min etc.
* **df.plot.line()-**It is going plot a graph for the specified column
  + If we want plot graph for all the columns then we can use the following syntax:
    - df.plot.line(subplots= True)

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

**MATPLOTLIB:**

* Matplotlib is a versatile Python library used for data visualisation in fields like data analysis, scientific computing, and machine learning.
* It offers various plot types, including line plots, scatter plots, and histograms, making it suitable for exploratory data analysis.
* Matplotlib is also employed in machine learning projects for visualising model performance and feature distributions.
* Additionally, it supports geospatial data visualisation and interactive plotting capabilities, enabling users to create maps and dynamic visualisations.
* Overall, its flexibility, extensive functionality, and ease of use make it a popular choice.
* To import matplotlib library the syntax is:
  + **"import matplotlib.pyplot as plt” .**
* FUNCTIONS IN MATPLOTLIB:
* **plt.plot(parameters)-**It is used to plot for the given information as parameters.
  + It also accepts the “colour” parameter where we can give the colour of the graph.
* **plt.title(“title\_name”)-**This function is used to give a title to graph
  + The title is passed as a parameter in the quotes.
* **plt.legend(loc=”location”)-**There are more parameters to plot function such as labels where we can label the things for which we are plotting a graph.
  + This function works based on the labels.Here we can locate where we want to place the labels and the location is specified to the **“loc” .**
* **plt.show()-**This function is used to show our graph.
* **plt.subplot()-**This function gives the separate graph,without this function all the graphs will be shown in a single graph
* **plt.pie()-**This function is used to give pie chart based on the information given by us as the parameters
* **plt.xlabel()-**It is used to give the name or label to the x axis.
* **plt.ylabel()-**It is used to give the name or label to the y axis.

**SEABORN:**

* Seaborn, a Python data visualization library, simplifies the creation of statistical graphics for data exploration and analysis.
* It offers high-level interfaces for visualizing relationships between variables, handling categorical data, and creating heatmaps and matrices.
* It provides tools for visualising time series data and customizing plot aesthetics.
* Seamlessly integrating with Pandas, Seaborn enables users to leverage its visualisation capabilities directly on DataFrame objects.
* To import seaborn the syntax is:
  + Import seaborn as sns.
* Seaborn has 14 -15 datasets in it as default.They are:

1. Anscombe: Anscombe's quartet dataset.

2. attention: Response times in a psychological experiment.

3. brain\_networks: Coordinates of networks in the human brain.

4. car\_crashes: Insurance data about car crashes.

5. diamonds: Characteristics of diamonds.

6. dots: Lateralized response times in a psychological experiment.

7. exercise: Measurements of exercise patterns.

8. flights: Data about flights.

9. fmri: Functional magnetic resonance imaging (fMRI) data.

10. gammas: Brain activity during exposure to gamma rays.

11. iris: Iris flower data.

12. mpg: Miles per gallon (MPG) and various car attributes.

13. planets: Exoplanets data.

14. tips: Restaurant tips data.

* To load these default datasets the command is as follows:
  + **var\_name=sns.load\_dataset(“dataset name”)**
* FUNCTIONS IN SEABORN:
* Seaborn is mainly used for plotting which have it consists many types of plotting functions.They are:
  + **sns.scatterplot()**
  + **sns.jointplot()**
  + **sns.lineplot()**
  + **sns.violinplot()**
  + **sns.lmplot()**
  + **sns.barplot()**
  + **sns.boxplot()**
  + **sns.countplot()**
  + **sns.FacetGrid()**
  + **sns.heatmap()**

All these functions gives different plotting style graphs as output.

**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:**

It's 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 the 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 functions 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 the 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, reduces 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-5(28/02/2024)**

**LINEAR REGRESSION:**

* It learns from labeled datasets and maps the data points to the most optimized linear functions.
* These points can be used for prediction on new datasets.
* We have two variables,they are dependent and independent.

The formulae is y=a+bx.To find

1. We have to calculate the mean for both the dependent and independent.
2. Find the differences between x point and x’(mean x).
3. Find the differences between y point and y’(mean y).
4. Find the total sum of squares of (x-x’).
5. Find the total sum of products of (x-x’) and (y-y’).

* b=sum of product of (x-x’) and (y-y’).

total sum of squares of (x-x’)

* Now with the help of **b** we can find **a** by replacing **x** with **x’** and **y** with **y’** in **y=ax+b.**
* The main two functions in this linear regression are fit() and predict().

**LOGISTIC REGRESSION:**

* It is used for binary classification and is learned from supervised algorithms.
* But the only difference is this will actually tell us which class our prediction site is in.

SIGMOID FUNCTION

* When the model is confident then it shows a narrow decision boundary
* When the model is not confident it shows a wide decision boundary.

**DECISION TREE:**

* Decision trees in machine learning provide an effective method for making decisions.
* This is because they lay out the problem and all the possible outcomes.
* They have **nodes** and **leafs.**
  + **Nodes:** They have both the true and false branches.
  + **leaf:** It has either true or false in other words it shows us the result and where our outcome sits.
* Decision trees recurrently(continuously) split the data until it gets pure nodes(leaf).

**DISADVANTAGES:**

* It has no accuracy.
* It won't get to a conclusion until it recurrently splits the data which wastes time.

**RANDOM FOREST:**

* Collection of many decision trees is called Random Forest.
* We can frequently find **bootstrapping** and **aggregation.**
* Random forest creates a new dataset from the existing or the original dataset.
  + This process is called **“bootstrapping”.**
* This gives the more accurate answers when compared to the decision tree.

* CONDITIONS:
  + Select rows that are different and cover all the rows.
  + Select features that are different and cover all features.
  + Selecting the same data(rows) and same features have no use.

**Hypothesis Testing:**

1. **T-test :**

It is a measure of how many standard deviations and sample mean are away from the hypothesized population mean,related to the variability in the sample.

It is calculated as the difference between sample mean and the population mean divided by the standard error of sample mean.

A small p value (typically less than 0.05) indicates that the observed data is unlike the under the null hypothesis is true leading to the rejection of the null hypothesis in favor of the alternative hypothesis.

Conversely a large t value suggests that the observed data is under the null hypothesis leading to a failure to reject the null hypothesis.

**Hypothesis Testing:**

If the p value is predetermined significant value (suggested 0.05) it is typically interpreted as sufficient evidence to reject the hypothesis

If the p value is greater than the significance level there is not enough evidence to reject the null hypothesis

Together t\_static and p\_value it is determining whether the observed sample data provides enough evidence to support and claim of hypothesis about the population.