8/10/2025 **PYTHON**

Libraries:

- They contain the modules and functions which perform a specific task.
- Types:
- **Pandas**: Used for data manipulation
- **NumPy**: Numerical python used for multi-dimensional array and mathematical operations.
- **Statistics:** For all kind of statistical operations.
- **Matplotlib, seaborn, plotly:** used for data visualization.
- **Sklearn(scikit learn):** used for all ML operations.
- **Keras:** used for deep learning, helpful for neural n/w.
- **Tensorflow:** used for deep learning[developed by google].
- **Pytorch:** used for deep leraning[developed by facebook].

- We have to import pandas: import pandas as pd.
- To overcome module error: pip install module_name.
- To avoid warnings like 1 pg warnings: import warnings

  warnings.filterwarnings('ignore')

- Pandas and ML library supports tabular data.
- In pandas fetching data is we use 'read'.

  Eg: import pandas as pd

  D1=pd.read_excel

- Text file is also in csv format so read as csv file.
- tsv is also in csv format just a separator is tab space.

  Eg: d2=pd.read_csv(r"c:\users\hp\download\chiptole.tsv", sep='\t')

- To see how many rows and cols are present

  Eg: d2.shape

- To access the element: d2.shape[0].
- To get no. of records from top: d2.head() [default it returns 1st 5].
- To get no, of records from bottom: d2.tail() [default it returns 5].
- To get random records: d2.sample()

- To print only the column: d2.columns
- To check in each col whether null values are present: d2.isna()
- To count no. of null values: d2.isna().sum()
- In pandas we have only 3 datatypes: int32/int64,float32/float64,object[for str]
- To check the datatype for col: d2.dtypes
- To get all info above together: d2.info()
- To find total null values: [to entries-null value]:32-4=28(null values).
- To fetch the 1st col: d2.manufacturer or d2[manufacturer].
- If we get row index like values then its series frame.
- Series is always 1D.
- DataFrame is always 2D[i.e.,row and col]: d2[[‘manufacturer’]]
- To get series manually: pd.Series([‘a’,’b’,’c’,’d’])
- To get dataframe: dict={‘name’:[‘a’,’b’.’c’],’age’:[2,3,4]}

  pd.DataFrame(dict)

- To drop the col: d2.drop(‘carb’,axis=1)
- To permanently delete particular col for some particular file but it wont delete in main file,that time you store back to var.

  Eg: d2=d2.drop(‘carb’,axis=1)

- Iloc and loc is a dataframe slicing method.
- Synatx: var iloc[rstp:rspp:rinc/rdec,cstp:cspp:cinc/cdec]
- To extract all rows,1st 5 cols:d2.iloc[:32:1,0:5:1]
- To extract complete table: d2.iloc[:,:]
- To add new col to dataframe: d2[‘temp’]=d2[‘mpg’]*d2[‘wt’]
- To update col: d2[‘cyl’]=d2[‘cyl’]+10
- To permanently rename col: d2.rename({‘drat’:’dratio’},axis=1,inplace=True)
- To sort in ascending order: d2.sort_values(‘mpg’,inplace=True)
- To get only particular col in asc order: d2[‘mpg’].sort_values()
- To sort in descending order: d2.sort_values(‘mpg’,ascending=False)
- To find mean,median,mode,max,min,count of particular col: d2.mpg.mean(),d2.mpg.median(),d2.mpg.mode(),d2.mpg.max(),d2.mpg.min(),d2.mpg.count()
- To describe the above full: d2.mpg,describe()
- To display all categorical data: d2.describe(include=’all’)

- To find unique values in particular col: d2.carb.nunique()
- To find no. of records for particular col: d2.carb.value_counts()
- To join multiple table: order should be same and col names should be same.
- To merge the tables: data=pd.concat([sale1,sale2,sale3])
- To not consider existing table index,we have to create new index: data=pd.concat([sale1,sale2,sale3],ignore_index=True)
- Any data stored in continuous date format or which is dependent on other predictions is known as ‘Time series format’.