Pandas Notes

Info pandas:

Pandas is an open-source Python library that provides high-performance, easy-to-use data structures, and data analysis tools. It is widely used for data manipulation, cleaning, and analysis in fields like data science, machine learning, and statistical research.

**Key Features of Pandas:**

1. **Data Structures**:
   * **Series**: A one-dimensional labeled array, similar to a list or a column in a spreadsheet.
   * **DataFrame**: A two-dimensional labeled data structure, similar to a table in a relational database or an Excel spreadsheet.
   * **Panel (deprecated)**: A three-dimensional data structure (no longer actively supported).
2. **Data Handling**:
   * Efficient handling of **missing data**.
   * Support for **heterogeneous data types**.
3. **Data Manipulation**:
   * Merging, joining, reshaping, and pivoting datasets.
   * Filtering, sorting, and grouping data with ease.
   * Built-in support for **time-series data**.
4. **Integration**:
   * Works well with other libraries like NumPy, Matplotlib, and Scikit-learn.
   * Read/write support for various file formats like CSV, Excel, SQL databases, JSON, etc.
5. **Vectorized Operations**:
   * Performs fast, element-wise operations on data, avoiding explicit Python loops.
6. **Visualization**:
   * Provides simple integration with plotting libraries for data visualization.
7. When reading excel file first excute this command (pip install openpyxl )then read excel file
8. When reading csv file (pd.read\_csv(“file.csv”))

Handling null values:

3)To get info about data use:

df.info()

1. Numerical description df.describe()
2. Finding duplicated value in dataset

df[“colname”].duplicated().sum()

1. To remove duplicated make sure don’t remove anything remove only like id of any data set :

df.drop\_duplicates("id")

1. The dropna() function in **Pandas** is used to remove missing values (NaN) from a DataFrame or Series. This is useful for cleaning datasets by dropping rows or columns that contain null values.

**Syntax:**

DataFrame.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)

**Parameters:**

**axis** (default 0): Determines whether to drop rows or columns.

0 or 'index': Drop rows with missing values.

1 or 'columns': Drop columns with missing values.

**how** (default 'any'): Specifies how to drop rows/columns.

'any': Drop if **any** value is missing.

'all': Drop only if **all** values are missing.

**thresh**: Specifies the minimum number of **non-null** values required to keep a row or column. Overrides how if specified.

**subset**: A list of column labels to consider when dropping rows. Only the specified columns are checked for missing values.

**inplace** (default False): Whether to perform the operation in place or return a new DataFrame.

True: Modify the DataFrame in place.

False: Return a new DataFrame.

1. Replace():

The replace() function in **Pandas** is used to replace values in a DataFrame or Series with specified values. It is a flexible and powerful way to clean or transform data by substituting specific values or patterns.

**Synatx:dataframe.replace(what to replace(NaN),what add instead of it(0)**)

# Replace value 2 with 20

df\_replaced = df.replace(2, 20)

print(df\_replaced)

1. Fillna():

**method**: Specifies the method to use for filling NaN. Options are:

* 'ffill' or 'pad': Forward fill (propagate last valid value forward).
* 'bfill' or 'backfill': Backward fill (propagate next valid value backward).

Dataframe.fillna(method=’ffill/bfill’)

Columns transformation in pandas

The loc function in **Pandas** allows you to perform **label-based indexing** and is a powerful tool for transforming specific columns of a DataFrame. You can use it to update, manipulate, or transform columns based on conditions or apply transformations across the entire column.

**Syntax for loc with Column Transformation:**

DataFrame.loc[rows, column] = value\_or\_transformation

* **rows**: Specify the row(s) to operate on.
* **column**: Specify the column(s) to transform.
* **value\_or\_transformation**: Provide a value or a transformation (e.g., a function).
* **Eg**:

# Replace values in column 'A' that are greater than 15 with 99

df.loc[df['A'] > 15, 'A'] = 99

print(df)