Pandas

Pandas

- Python Pandas is defined as an open-source library that provides high-performance data manipulation in Python.
- The name of Pandas is derived from the word Panel Data, which means an Econometrics from Multidimensional data.
- It is used for data analysis in Python and developed by Wes McKinney in 2008.
- It has functions for analyzing, cleaning, exploring, and manipulating data.

Installation of Pandas

- Pip install pandas
- Once Pandas is installed, import it in your applications by adding the import keyword
- Import pandas
- Pandas is usually imported under the pd alias.
- Import pandas as pd

Python Pandas Data Structure

- Pandas provides two data structures for processing the data, i.e., Series and DataFrame
- Series: one-dimensional array that is capable of storing various data types. The row labels of series are called the index.
- DataFrame is defined as a standard way to store data and has two different indexes, i.e., row index and column index.

Pandas Series

- Pandas Series can be defined as a one-dimensional array that is capable of storing various data types.
- <series object> = pandas.Series()

Series object attributes

- Series.index Defines the index of the Series.
- Series.shape It returns a tuple of shape of the data.
- Series.dtype It returns the data type of the data.
- Series.size It returns the size of the data.
- Series.empty It returns True if Series object is empty, otherwise returns false.
- Series.hasnans It returns True if there are any NaN values, otherwise returns false.
- Series.nbytes It returns the number of bytes in the data.
- Series.ndim It returns the number of dimensions in the data.

Series Functions

- Len(series)
- Series.count()
- Series.std()
- Max()
- Unique()
- Value_counts()

Pandas DataFrames

- A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

```
data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

DataFrame functions

- Aggregate functions
- Assignining Rows and columns-assign()
- Count()
- Cut()
- Describe()
- Drop_duplicates()
- Head()
- Mean()

Dataframe functions

- Mean()
- Rename()-to rename a column

Data Wrangling in Python

Pandas Read CSV

- A simple way to store big data sets is to use CSV files (comma separated files).
- CSV files contains plain text and is a well know format that can be read by everyone including Pandas.
- large DataFrame with many rows, Pandas will only return the first 5 rows, and the last 5 rows
- df = pd.read_csv('data.csv')
- print(df)

Viewing Data

- use to_string() to print the entire DataFrame.
- df = pd.read_csv('data.csv')
- print(df.to_string())
- head() method returns the headers and a specified number of rows, starting from the top.
- print(df.head(10))
- tail() method for viewing the last rows of the DataFrame.

Pandas - Cleaning Data

- to change the original DataFrame, use the inplace = True argument
- `df.dropna(inplace = True)`
- The fillna() method allows us to replace empty cells with a value
- df.fillna(130, inplace = True)
- To only replace empty values for one column, specify the column name for the DataFrame
- df["Calories"].fillna(130, inplace = True)

Replace Using Mean, Median, or Mode

- x = df["Calories"].mean()
- x = df["Calories"].median()
- x = df["Calories"].mode()[0]
- df["Calories"].fillna(x, inplace = True)

Pandas - Removing Duplicates

- To discover duplicates, we can use the duplicated() method.
- print(df.duplicated())
- To remove duplicates, use the drop_duplicates() method
- df.drop_duplicates(inplace = True)