Pandas Practice

It is an open-source dataset and you can download it from

https://github.com/dsrscientist/Python-\_ML-\_Extras\_Repo/blob/main/IMDB-Movie-Data.csv

We will read the data from the .csv file and perform the following basic operations on movies data

1. Load the IMDb Dataset and read
2. View the dataset
3. Understand some basic information about the dataset and Inspect the dataframe Inspect the dataframe's columns, shapes, variable types etc.
4. Data Selection – Indexing and Slicing data
5. Data Selection – Based on Conditional filtering
6. Groupby operations
7. Sorting operation
8. Dealing with missing values
9. Dropping columns and null values
10. Apply( ) function

Solutions

Load the dataset

*import numpy as np*

*iimport pandas as pd*

*df = pd.read\_csv('IMDB-Movie-Data.csv')*

**2. View the dataset**

*df.head(10)*

**3. Understand some basic information about the dataset and Inspect the dataframe Inspect the dataframe's columns, shapes, variable types etc.**

*df.columns*

*type(df)*

*df.dtypes*

*df.shape*

*df.size*

*df.ndim*

*df.values*

*df1 = df.values*

*type(df1)*

*df.describe()*

**4. Data Selection – Indexing and Slicing data**

*df.iloc[0]*

*df[1:20]*

*df[['Rating','Votes']].agg(['min','max','mean'])*

**5. Data Selection – Based on Conditional filtering**

*df.filter(items=['Rank', 'Votes'])*

*df['Rating']>7*

*top\_rank = df[df["Rating"] > 8.0]["Title"].count()*

*print(top\_rank)*

**6. Groupby operations**

*df2 = df.groupby('Genre')*

*df2.mean()*

*df[df['Rating']>7].groupby('Genre')[['Rating']].count()*

*top\_movie = df[df["Rating"] > 8.0]*

*top\_movie.groupby(["Title"])["Votes"].mean()*

**7. Sorting operation**

*x = df.sort\_values(by='Rating')*

*x.head(10)*

most\_votes = df.groupby(["Votes"]).mean()

most\_votes.sort\_values(by = ["Votes"], ascending = False).head()

**8. Dealing with missing values**

*df.isnull().sum()*

**9. Dropping columns and null values**

*df.dropna()*

*x = df.drop(['Metascore'], axis='columns', inplace=True)*

**10. Apply( ) functions**

*rank = df.apply(lambda n: n\*5)*

*print(rank.head())*