In [5]:

```
'''
Question 1:
Create a CSV file called “Movies.csv” with details of 10 movies- Movie Name,
a. Read CSV file into a dataframe and find the movie with the highest rating
b. Write the details of all “Hindi movies into a file “HindiMovies.csv”.
'''
import pandas as pd
import numpy as np

data = {
    "Movie Name": ["Inception", "3 Idiots", "Parasite", "Interstellar", "Dar
    "Language": ["English", "Hindi", "Korean", "English", "Hindi", "English"
    "Genre": ["Sci-Fi", "Comedy", "Thriller", "Sci-Fi", "Drama", "Action", "
    "Rating": [8.8, 8.4, 8.6, 8.6, 8.4, 9.0, 8.1, 8.4, 8.1, 9.2],
    "Review": [
        "Mind-bending thriller", "Inspirational and funny", "Masterpiece of
        "Heartwarming sports drama", "Brilliantly dark and intense", "Though
        "Epic historical drama", "Timeless classic"
    ]
}

df = pd.DataFrame(data)
df.to_csv("Movies.csv", index=False)

df = pd.read_csv("Movies.csv")
df

highest_rated_movie = df.loc[df['Rating'].idxmax()]
print(f"Movie with highest rating\n\n{highest_rated_movie}")

hindi_movies = df[df["Language"] == "Hindi"]
hindi_movies.to_csv("HindiMovies.csv", index=False)
print("Hindi movies written to HindiMovies.csv")
df2 = pd.read_csv("HindiMovies.csv")
df2
```

```
Movie with highest rating

Movie Name       The Godfather
Language               English
Genre                    Crime
Rating                     9.2
Review        Timeless classic
Name: 9, dtype: object
Hindi movies written to HindiMovies.csv
```

Out[5]:

	Movie Name	Language	Genre	Rating	Review
0	3 Idiots	Hindi	Comedy	8.4	Inspirational and funny
1	Dangal	Hindi	Drama	8.4	Heartwarming sports drama
2	PK	Hindi	Comedy	8.1	Thought-provoking
3	Lagaan	Hindi	Sports	8.1	Epic historical drama

In [8]:

```
'''
Question 2:
For the CEREALS dataset, perform data preprocessing and answer the following
a. Create a table with the 5 number summary of all the numeric attributes.
b. For each of the numeric attributes (proteins upto vitamins) , identify an
c. Create a table with the 5 number summary of all the numeric attributes af
d. For each numeric attribute (proteins upto vitamins), identify and replace
e. Create a table with the 5 number summary of all the numeric attributes af
'''
import pandas as pd
import numpy as np

df = pd.read_excel("Cereals.xls")

initial_numeric_summary = df.describe().loc[['min', '25%', '50%', '75%', 'ma
print(f"Initial five-number summary of all numeric attributes\n\n{initial_nu

numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].mean())
print(f"\n\n{df}")

df.replace(-1, pd.NA, inplace=True)
df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].mean())

numeric_summary_missing_handling=df.describe().loc[['min', '25%', '50%', '75
print(f"Five-number summary of all numeric attributes after handling missing

for column in df.select_dtypes(include=['float64', 'int64']).columns:
  upp = df[column].quantile(0.95)
  low = df[column].quantile(0.05)
  med = df[column].median()
  df[column] = df[column].apply(lambda x: med if (x > upp or x < low) else x

numeric_summary_noisy_handling = df.describe().loc[['min', '25%', '50%', '75
print(f"Five-number summary after handling noisy values\n\n{numeric_summary_
```

Initial five-number summary of all numeric attributes

```
     calories  protein  fat  sodium  fiber  carbo  sugars  potass  vitamin
s  \
min      50.0      1.0  0.0     0.0   0.00   -1.0    -1.0    -1.0       0.
0
25%     100.0      2.0  0.0   132.5   0.75   12.0     3.0    40.0      25.
0
50%     110.0      2.5  1.0   180.0   1.75   14.5     7.0    90.0      25.
0
75%     110.0      3.0  2.0   212.5   3.00   17.0    11.0   120.0      25.
0
max     160.0      6.0  5.0   320.0  14.00   23.0    15.0   330.0     100.
0

     shelf  weight  cups     rating
min    1.0     0.5  0.25  18.042851
25%    1.0     1.0  0.67  32.932466
50%    2.0     1.0  0.75  40.253086
75%    3.0     1.0  1.00  50.780847
max    3.0     1.5  1.50  93.704912
```

	name	mfr	type	calories	protein	fat	sodium	fiber
0	100%_Natural_Bran	Q	C	120	3	5	15	2.0
1	All-Bran	K	C	70	4	1	260	9.0
2	All-Bran_with_Extra_Fiber	K	C	50	4	0	140	14.0
3	Almond_Delight	R	C	110	2	2	200	1.0
4	Apple_Cinnamon_Cheerios	G	C	110	2	2	180	1.5
..	...	..	...	...	...	...	...	...
71	Triples	G	C	110	2	1	250	0.0
72	Trix	G	C	110	1	1	140	0.0
73	Wheat_Chex	R	C	100	3	1	230	3.0
74	Wheaties	G	C	100	3	1	200	3.0
75	Wheaties_Honey_Gold	G	C	110	2	1	200	1.0

	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	8.0	8	135	0	3	1.0	1.00	33.983679
1	7.0	5	320	25	3	1.0	0.33	59.425505
2	8.0	0	330	25	3	1.0	0.50	93.704912
3	14.0	8	-1	25	3	1.0	0.75	34.384843
4	10.5	10	70	25	1	1.0	0.75	29.509541
..	...	...	...	...	...	...	...	...
71	21.0	3	60	25	3	1.0	0.75	39.106174
72	13.0	12	25	25	2	1.0	1.00	27.753301
73	17.0	3	115	25	1	1.0	0.67	49.787445
74	17.0	3	110	25	1	1.0	1.00	51.592193
75	16.0	8	60	25	1	1.0	0.75	36.187559

[76 rows x 16 columns]
Five-number summary of all numeric attributes after handling missing data

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	\
min	50.0	1.0	0.0	0.0	0.00	7.000000	0.0	15.00	
25%	100.0	2.0	0.0	132.5	0.75	12.000000	3.0	43.75	
50%	110.0	2.5	1.0	180.0	1.75	14.966667	7.0	90.00	
75%	110.0	3.0	2.0	212.5	3.00	17.000000	11.0	120.00	
max	160.0	6.0	5.0	320.0	14.00	23.000000	15.0	330.00	

	vitamins	shelf	weight	cups	rating
min	0.0	1.0	0.5	0.25	18.042851
25%	25.0	1.0	1.0	0.67	32.932466
50%	25.0	2.0	1.0	0.75	40.253086
75%	25.0	3.0	1.0	1.00	50.780847
max	100.0	3.0	1.5	1.50	93.704912

Five-number summary after handling noisy values

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	\
min	80.0	1.00	0.00	0.0	0.000	10.000000	0.00	25.0	
25%	100.0	2.00	0.00	132.5	0.750	13.000000	3.00	45.0	
50%	110.0	2.25	1.00	180.0	1.625	14.966667	7.00	90.0	
75%	110.0	3.00	1.25	202.5	3.000	17.000000	10.25	110.0	
max	140.0	4.00	3.00	280.0	5.000	21.000000	14.00	230.0	

	vitamins	shelf	weight	cups	rating
min	0.0	1.0	1.00	0.50	22.736446
25%	25.0	1.0	1.00	0.67	35.035544
50%	25.0	2.0	1.00	0.75	40.253086
75%	25.0	3.0	1.00	1.00	47.451796
max	100.0	3.0	1.33	1.00	64.533816