EDS

Theory Assignment: 01

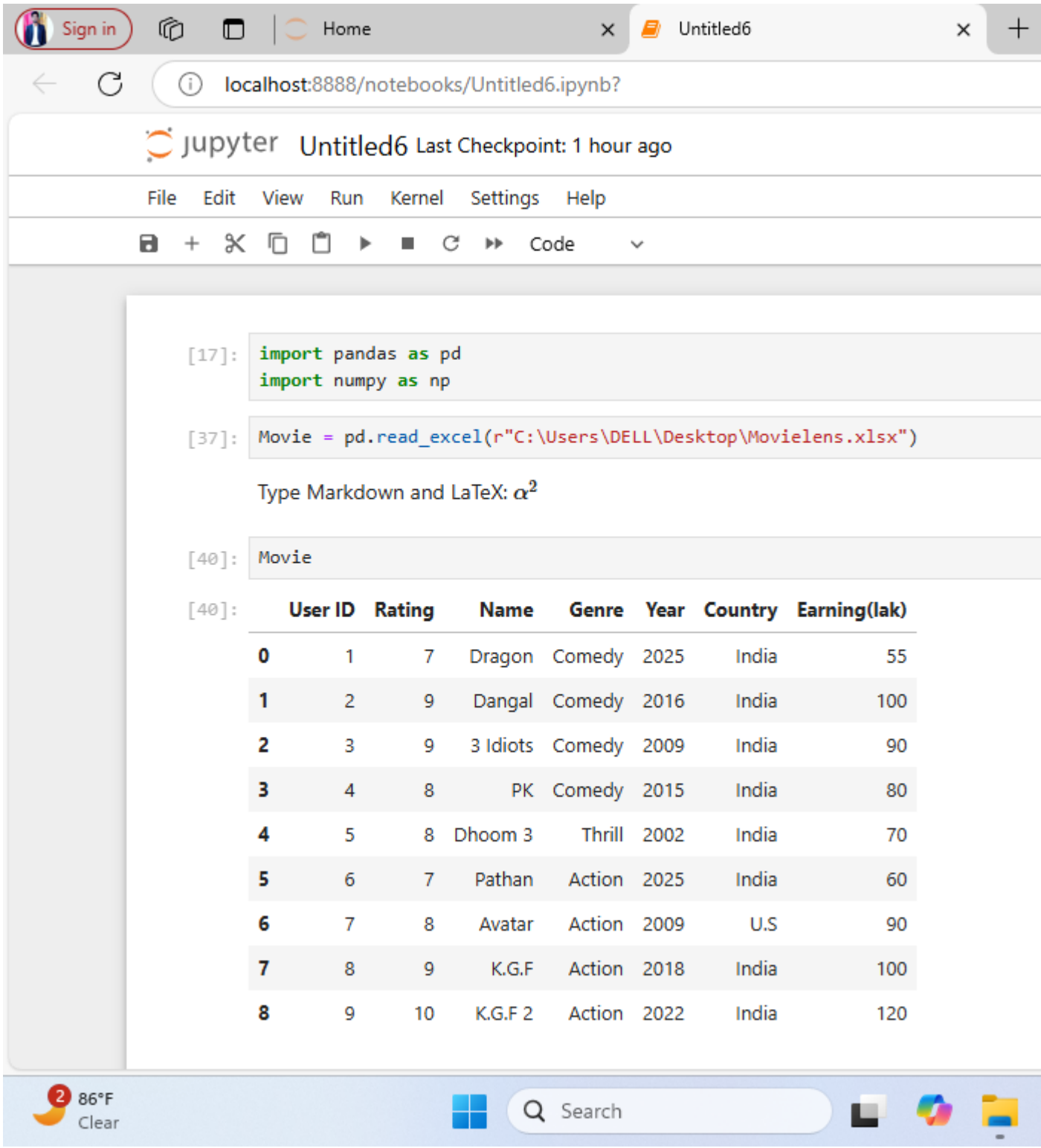
NAME: Rushikesh Shinde

PRN: 2024010800048

Roll no: CS6-75

Batch: C64

DATA SET: MOVIELENS latest datasets



	User ID	Rating	Name	Genre	Year	Country	Earning(lak)
0	1	7	Dragon	Comedy	2025	India	55
1	2	9	Dangal	Comedy	2016	India	100
2	3	9	3 Idiots	Comedy	2009	India	90
3	4	8	PK	Comedy	2015	India	80
4	5	8	Dhoom 3	Thrill	2002	India	70
5	6	7	Pathan	Action	2025	India	60
6	7	8	Avatar	Action	2009	U.S	90
7	8	9	K.G.F	Action	2018	India	100
8	9	10	K.G.F 2	Action	2022	India	120

Q1: Find the total number of unique players.

```
print(Movie['Name'].nunique())
```

```
9
```

Q2: Display first 5 rows

```
[44]: Movie.head(5)
```

[44]:

	User ID	Rating	Name	Genre	Year	Country	Earning(lak)
0	1	7	Dragon	Comedy	2025	India	55
1	2	9	Dangal	Comedy	2016	India	100
2	3	9	3 Idiots	Comedy	2009	India	90
3	4	8	PK	Comedy	2015	India	80
4	5	8	Dhoom 3	Thrill	2002	India	70

Q3: Display the last 4 rows

```
]: Movie.tail(4)
```

]:

	User ID	Rating	Name	Genre	Year	Country	Earning(lak)
5	6	7	Pathan	Action	2025	India	60
6	7	8	Avatar	Action	2009	U.S	90
7	8	9	K.G.F	Action	2018	India	100
8	9	10	K.G.F 2	Action	2022	India	120

Q4: Description about full dataset

```
]: print(Movie.describe())
```

```
        User ID     Rating         Year  Earning(lak)
count  9.000000   9.000000     9.000000      9.000000
mean   5.000000   8.333333  2015.666667     85.000000
std    2.738613   1.000000     7.874008     20.916501
min    1.000000   7.000000  2002.000000     55.000000
25%    3.000000   8.000000  2009.000000     70.000000
50%    5.000000   8.000000  2016.000000     90.000000
75%    7.000000   9.000000  2022.000000    100.000000
max    9.000000  10.000000  2025.000000    120.000000
```

Q5: Rating less than 9 and year greater than 2002

```
: young_movies=Movie[(Movie['Rating']<9)&(Movie['Year']>2002)]
```

```
: young_movies
```

```
:
```

	User ID	Rating	Name	Genre	Year	Country	Earning(lak)
0	1	7	Dragon	Comedy	2025	India	55
3	4	8	PK	Comedy	2015	India	80
5	6	7	Pathan	Action	2025	India	60
6	7	8	Avatar	Action	2009	U.S	90

Q6: Top 5 earning movies

```
top_movies = Movie.sort_values('Earning(lak)',ascending=False).head(5)
```

```
top_movies
```

	User ID	Rating	Name	Genre	Year	Country	Earning(lak)
8	9	10	K.G.F 2	Action	2022	India	120
1	2	9	Dangal	Comedy	2016	India	100
7	8	9	K.G.F	Action	2018	India	100
2	3	9	3 Idiots	Comedy	2009	India	90
6	7	8	Avatar	Action	2009	U.S	90

Q7: Rating less than 9 and country = India

```
indian_movies=Movie[(Movie['Rating']<9)&(Movie['Country']=='India')]
print(indian_movies)
```

```
   User ID  Rating     Name   Genre  Year Country  Earning(lak)
0        1       7   Dragon  Comedy  2025   India            55
3        4       8       PK  Comedy  2015   India            80
4        5       8  Dhoom 3  Thrill  2002   India            70
5        6       7   Pathan  Action  2025   India            60
```

Q8: Movie with earning greater than 50lak

```
hig_value = Movie[Movie['Earning(lak)'] > 50]
print(hig_value)
```

```
   User ID  Rating      Name   Genre  Year Country  Earning(lak)
0        1       7    Dragon  Comedy  2025   India            55
1        2       9    Dangal  Comedy  2016   India           100
2        3       9  3 Idiots  Comedy  2009   India            90
3        4       8        PK  Comedy  2015   India            80
4        5       8   Dhoom 3  Thrill  2002   India            70
5        6       7    Pathan  Action  2025   India            60
6        7       8    Avatar  Action  2009     U.S            90
7        8       9     K.G.F  Action  2018   India           100
8        9      10   K.G.F 2  Action  2022   India           120
```

Q9: correlation b/w rating and earning

```
print(Movie[['Rating','Earning(lak)']].corr())
```

```
                Rating  Earning(lak)
Rating        1.000000      0.956183
Earning(lak)  0.956183      1.000000
```

Q10: Check missing values(NULL values)

```
print(Movie.isnull().sum())
```

```
User ID         0
Rating          0
Name            0
Genre           0
Year            0
Country         0
Earning(lak)    0
dtype: int64
```

Q11: top 8 movies in sequence

```
top_mov = Movie['Name'].head(8)
```

```
print(top_mov)
```

```
0      Dragon
1      Dangal
2    3 Idiots
3          PK
4     Dhoom 3
5      Pathan
6      Avatar
7       K.G.F
Name: Name, dtype: object
```

Q12: best movie list(earning >50)

```
best_mov = Movie[Movie['Earning(lak)']>50]
print(best_mov)
```

	User ID	Rating	Name	Genre	Year	Country	Earning(lak)
0	1	7	Dragon	Comedy	2025	India	55
1	2	9	Dangal	Comedy	2016	India	100
2	3	9	3 Idiots	Comedy	2009	India	90
3	4	8	PK	Comedy	2015	India	80
4	5	8	Dhoom 3	Thrill	2002	India	70
5	6	7	Pathan	Action	2025	India	60
6	7	8	Avatar	Action	2009	U.S	90
7	8	9	K.G.F	Action	2018	India	100
8	9	10	K.G.F 2	Action	2022	India	120

Q13: Find mean, median, and standard deviation of Ratings

```
import numpy as np

Rating = Movie['Rating'].to_numpy()
print('Mean Rating :',np.mean(Rating))
print('Median Rating :',np.median(Rating))
print('STD Deviation :',np.std(Rating))
```

```
Mean Rating : 8.333333333333334
Median Rating : 8.0
STD Deviation : 0.9428090415820634
```

Q14: Movies name and rating got max rating in the table

```
over = Movie['Rating'].to_numpy()
max_index = np.argmax(over)
print(Movie.iloc[max_index][['Name','Rating']])
```

```
Name      K.G.F 2
Rating         10
Name: 8, dtype: object
```

Q15: find the standard deviation of rating grouped by country

```
grouped = Movie.groupby('Country')['Rating'].apply(lambda x: np.std(x.to_numpy()))
print(grouped)
```

```
Country
India    0.992157
U.S      0.000000
Name: Rating, dtype: float64
```

Q16: Correlation b/w earning (Lak) and rating

```
correlation = np.corrcoef(Movie['Rating'].to_numpy(), Movie['Earning(lak)'].to_numpy())[0,1]
print('correlation between Rating and Earning(lak) : ',correlation)
```

```
correlation between Rating and Earning(lak) :  0.9561828874675147
```

Q17: Max rating of the movie from the table

```
rating = Movie['Rating'].to_numpy()
print(np.max(rating))
```

```
10
```

Q18: Mean of Rating

```
value = Movie['Rating'].to_numpy()
print(np.mean(value))
```

```
8.333333333333334
```

Q19: Number of movies with Rating 9

```
count_9 = np.sum(Rating == 9)
print('Movie rating 9 : ',count_9)
```

```
Movie rating 9 :  3
```

Q20: Median of the Rating

```
print(np.median(Rating))
```

```
8.0
```

Q21: Datatypes of all columns

```
Movie.dtypes
```

```
User ID          int64
Rating           int64
Name            object
Genre           object
Year             int64
Country         object
Earning(lak)     int64
dtype: object
```