Project Description (Movielens Case Study):

“””

DESCRIPTION

Background of Problem Statement:

The GroupLens Research Project is a research group in the Department of Computer Science and Engineering at the University of Minnesota. Members of the GroupLens Research Project are involved in many research projects related to the fields of information filtering, collaborative filtering, and recommender systems. The project is led by professors John Riedl and Joseph Konstan. The project began to explore automated collaborative filtering in 1992 but is most well known for its worldwide trial of an automated collaborative filtering system for Usenet news in 1996. Since then the project has expanded its scope to research overall information by filtering solutions, integrating into content-based methods, as well as, improving current collaborative filtering technology.

Problem Objective:

Here, we ask you to perform the analysis using the Exploratory Data Analysis technique. You need to find features affecting the ratings of any particular movie and build a model to predict the movie ratings.

“””

Source Code of the Project:

```
#importing libraries


import pandas as pd

import warnings

warnings.filterwarnings('ignore')

import string

import numpy as np

from nltk.corpus import stopwords

from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split
```

```
from lightgbm import LGBMClassifier

from sklearn.linear_model import LinearRegression

import seaborn as sns

import matplotlib.pyplot as plt


#importing the datasets


movie_dataset=input("enter the path for movie dataset to import :")

movies_dataset=movie_dataset.replace("\\",'/')

movies=pd.read_csv(movies_dataset,delimiter='::',names=['MovieID','Title','Genere'])

rating_dataset=input("enter the path for rating dataset to import :")

ratings_dataset=rating_dataset.replace("\\",'/')

rating=pd.read_csv(ratings_dataset,delimiter='::',names=['UserID','MovieID','Rating','Timestamp'])

user_dataset=input("enter the path for user dataset to import :")

users_dataset=user_dataset.replace("\\",'/')

users=pd.read_csv(users_dataset,delimiter='::',names=['UserID','Gender','Age','Occupation','Zip-code'])


#Merging datasets for creating Master Dataset


merge_df=pd.merge(movies,rating,on='MovieID')

master_data=pd.merge(merge_df,users,on='UserID')

Master_Data=master_data.filter(['MovieID','Title','UserID','Age','Gender','Occupation','Rating'])

print("New Master_Data dataset with the columns('MovieID','Title','UserID','Age','Gender','Occupation','Rating') :")

print("_____
_____")
```

```
print(Master_Data)

#Visual Representations

sns.distplot(Master_Data[['Age']])
plt.title("User Age Distribution")
plt.show()
toy_story=Master_Data['Rating'].where(Master_Data['Title']=='Toy Story (1995)')
sns.distplot(toy_story)
plt.title("User Rating of the movie Toy Story")
plt.show()
top_rate=Master_Data.sort_values('Rating')
top_movies=top_rate.filter(['MovieID','Title','Rating'])
print("Top 25 movies by viewership rating :")
print("_____")
print(top_movies)
userid_2696=Master_Data['Rating'].where(Master_Data['UserID']==2696)
rating_2696=userid_2696.dropna()
sns.displot(rating_2696)
plt.title("Ratings for all the movies reviewed by a user of userid=2696")
plt.show()

#Feature Engineering Tasks:

generes=movies.Genere.tolist()
generes_list=[]
x=0
```

```
while(x<len(generes)):
  txt=generes[x].split('|')
  generes_list+=txt
  x=x+1
unique_generes=[]
for x in generes_list:
  if x not in unique_generes:
    unique_generes.append(x)
print("Unique Categories of Genres :")
print("_____")
print(unique_generes)
unique_genre=pd.DataFrame(unique_generes)
print("Number of Unique Categories of Genres in the document are :",len(unique_generes))
genere_each=pd.concat([Master_Data,movies['Genere'].str.get_dummies()],axis=1)
print("\nEach genre category with a one-hot encoding whether or not the movie belongs to that genre:")
print("_____
_____")
print(genere_each.fillna('0'))
factors=genere_each.drop(['MovieID','Title','UserID','Rating'],axis=1)
print("Factors affecting the ratings of any particular movie:")
print("_____")
print(factors.columns)
genere_encode_gender=pd.get_dummies(factors['Gender'])
x_feature=genere_encode_gender
y_target=Master_Data['Rating']
linreg=LGBMClassifier(boosting_type='gbdt',n_jobs=-1,objective='multiclass')
x_train,x_test,y_train,y_test=train_test_split(x_feature,y_target,random_state=1)
```

```
linreg.fit(x_train,y_train)

y_pred=linreg.predict(x_test)

print("Appropriate Model to predict the movie ratings:")

print("_____")

print("Accuracy Score : ", accuracy_score(y_pred,y_test)*100)
```

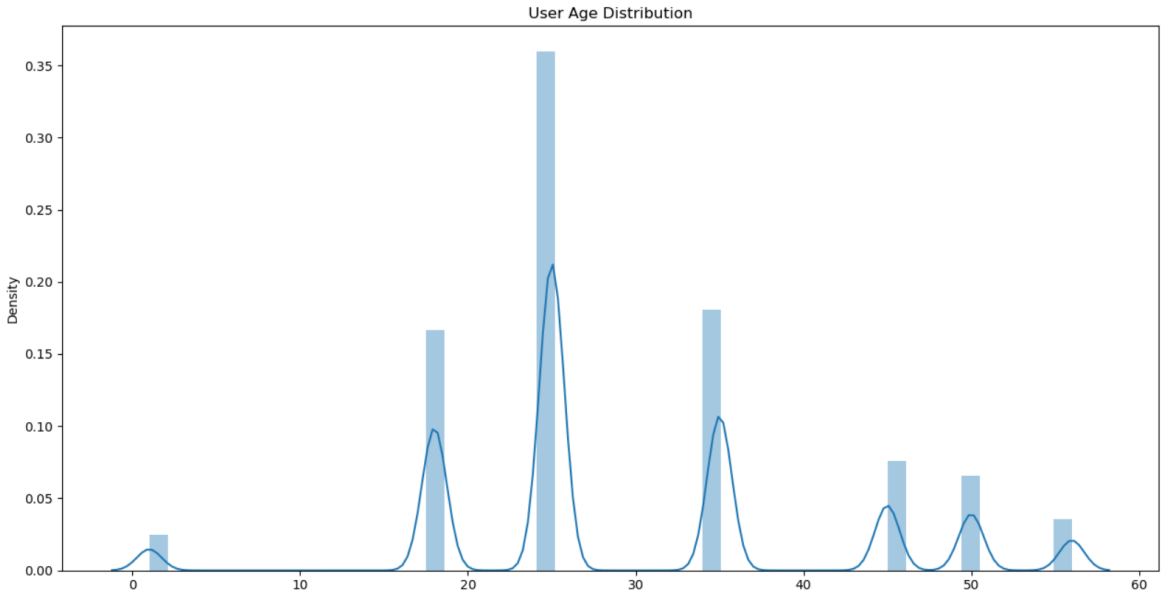
Screenshots of the Output:

Importing and Merging the datasets to create Master dataset:

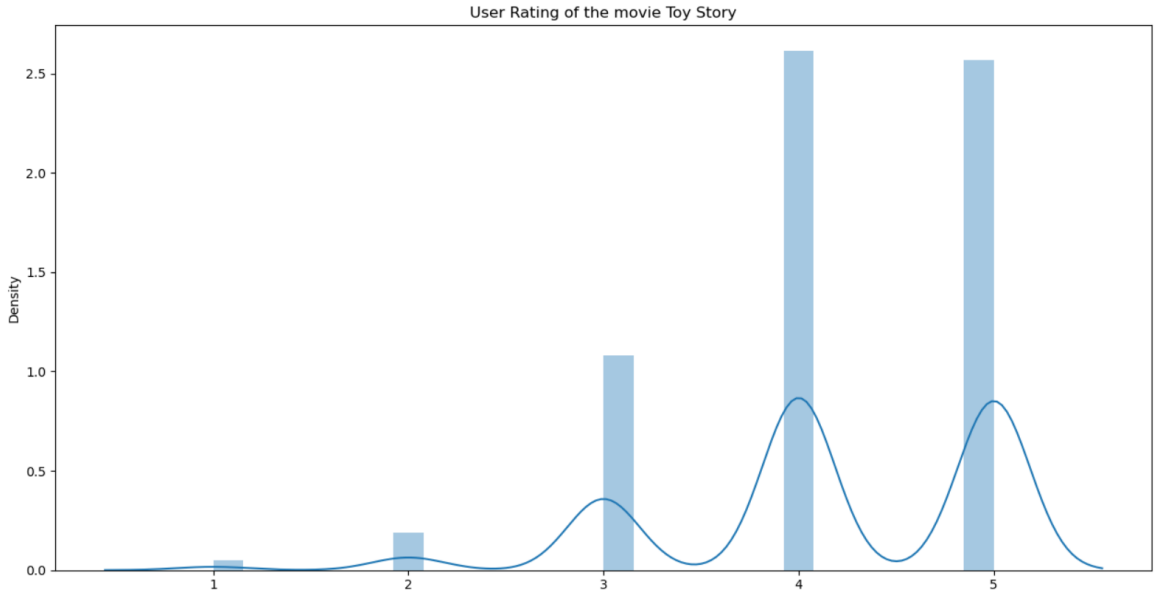
```
Administrator: Command Prompt

C:\Users\Swetha Thanjavur\Documents>python project1.py
enter the path for movie dataset to import :C:\Users\Swetha Thanjavur\Documents\movies.dat
enter the path for rating dataset to import :C:\Users\Swetha Thanjavur\Documents\ratings.dat
enter the path for user dataset to import :C:\Users\Swetha Thanjavur\Documents\users.dat
New Master_Data dataset with the columns('MovieID','Title','UserID','Age','Gender','Occupation','Rating') :
_____
         MovieID                                     Title  UserID  Age Gender  Occupation  Rating
0              1                          Toy Story (1995)       1    1      F          10       5
1             48                        Pocahontas (1995)       1    1      F          10       5
2            150                         Apollo 13 (1995)       1    1      F          10       5
3            260  Star Wars: Episode IV - A New Hope (1977)       1    1      F          10       4
4            527                  Schindler's List (1993)       1    1      F          10       5
...          ...                                      ...     ...  ...    ...         ...     ...
1000204     3513              Rules of Engagement (2000)    5727   25      M           4       4
1000205     3535                  American Psycho (2000)    5727   25      M           4       2
1000206     3536                Keeping the Faith (2000)    5727   25      M           4       5
1000207     3555                           U-571 (2000)    5727   25      M           4       3
1000208     3578                        Gladiator (2000)    5727   25      M           4       5
```

Graphical Representation of User Age Distribution:



Graphical representation of User Rating of Movie Toy Story:

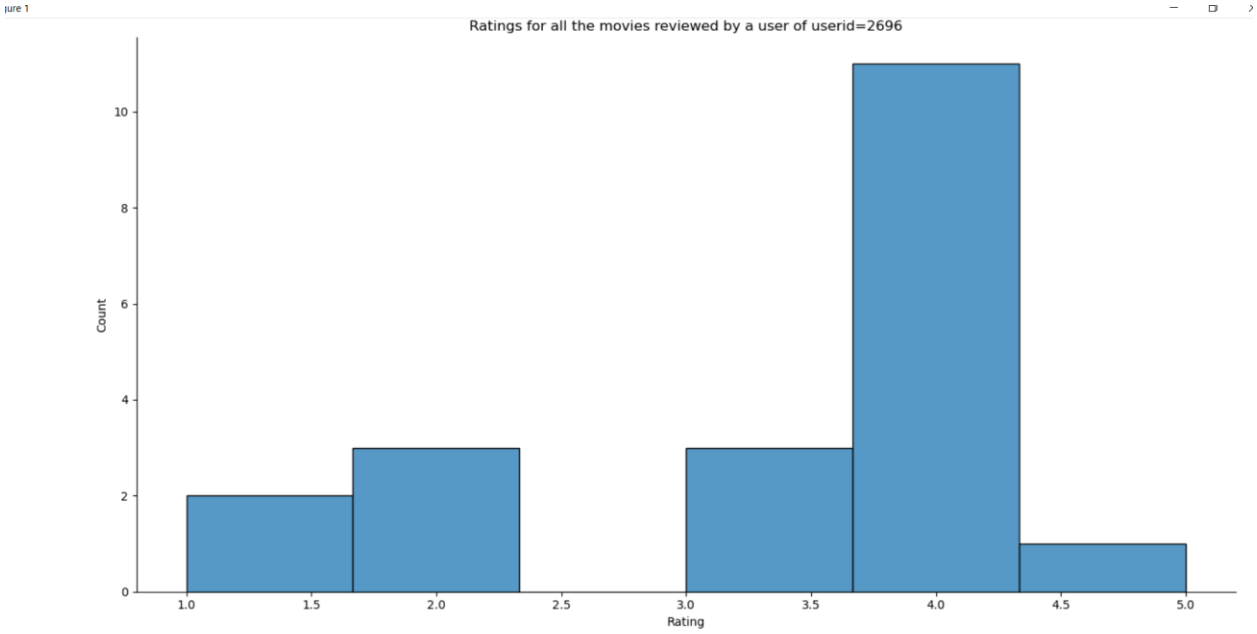


Top 25 movies by viewership Rating:

```
[1000209 rows x 7 columns]
Top 25 movies by viewership rating :
```

	MovieID	Title	Rating
857157	1680	Sliding Doors (1998)	1
575675	688	Operation Dumbo Drop (1995)	1
196709	473	In the Army Now (1994)	1
65101	2322	Soldier (1998)	1
65099	2298	Strangeland (1998)	1
...	...	...	...
562617	2414	Young Sherlock Holmes (1985)	5
562615	2396	Shakespeare in Love (1998)	5
562609	2300	Producers, The (1968)	5
562629	2819	Three Days of the Condor (1975)	5
1000208	3578	Gladiator (2000)	5

Graphical Representation of Ratings for all the movies reviewed by a user of userid=2696:



Unique Categories of Genres and one-hot encoding for each genre category:

```
Unique Categories of Genres :

['Animation', "Children's", 'Comedy', 'Adventure', 'Fantasy', 'Romance', 'Drama', 'Action', 'Crime', 'Thriller', 'Horror', 'Sci-Fi', 'Documentary', 'War', 'Musical', 'Mystery', 'Film-Noir', 'Western']
Number of Unique Categories of Genres in the document are : 18

Each genre category with a one-hot encoding whether or not the movie belongs to that genre:
```

	MovieID	Title	UserID	Age	Gender	Occupation	Rating	Action	Adventure	Animation	...	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thriller	War	Western
0	1	Toy Story (1995)	1	1	F	10	5	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	48	Pocahontas (1995)	1	1	F	10	5	0.0	1.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	150	Apollo 13 (1995)	1	1	F	10	5	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
3	260	Star Wars: Episode IV - A New Hope (1977)	1	1	F	10	4	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	527	Schindler's List (1993)	1	1	F	10	5	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1000204	3513	Rules of Engagement (2000)	5727	25	M	4	4	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1000205	3535	American Psycho (2000)	5727	25	M	4	2	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1000206	3536	Keeping the Faith (2000)	5727	25	M	4	5	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1000207	3555	U-571 (2000)	5727	25	M	4	3	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1000208	3578	Gladiator (2000)	5727	25	M	4	5	0	0	0	...	0	0	0	0	0	0	0	0	0	0

```
[1000209 rows x 25 columns]
```

Features affecting the ratings of any particular movie and appropriate model to predict the movie ratings:

```
Factors affecting the ratings of any particular movie:
_____________________________________________________________
Index(['Age', 'Gender', 'Occupation', 'Action', 'Adventure', 'Animation',
       'Children's', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy',
       'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi',
       'Thriller', 'War', 'Western'],
      dtype='object')
Appropriate Model to predict the movie ratings:
_____________________________________________________
Accuracy Score :  35.069765209775525
```