import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import warnings  
warnings.simplefilter('ignore')  
import seaborn  
from sklearn.model\_selection import train\_test\_split  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.metrics import confusion\_matrix, classification\_report,recall\_score,precision\_score  
  
'''  
1. Import the three datasets  
2. Create a new dataset [Master\_Data] with the following columns MovieID Title UserID Age Gender Occupation Rating. (Hint: (i) Merge two tables at a time. (ii) Merge the tables using two primary keys MovieID & UserId)  
3. Explore the datasets using visual representations (graphs or tables), also include your comments on the following:  
 3.a User Age Distribution  
 3.b User rating of the movie “Toy Story”  
 3.c Top 25 movies by viewership rating  
 3.d Find the ratings for all the movies reviewed by for a particular user of user id = 2696  
4. Feature Engineering:  
 Use column genres:  
 4.a Find out all the unique genres (Hint: split the data in column genre making a list and then process the data to find out only the unique categories of genres)  
 4.b Create a separate column for each genre category with a one-hot encoding ( 1 and 0) whether or not the movie belongs to that genre.   
 4.c Determine the features affecting the ratings of any particular movie.  
5. Develop an appropriate model to predict the movie rating  
'''  
  
movies = pd.read\_csv('movies.dat', sep='::', header=None, names=['movie\_id', 'movie\_name', 'genere'])  
# print(movies.shape)  
# print(movies.columns)  
ratings = pd.read\_csv('ratings.dat', sep='::', header=None, names=['user\_id','movie\_id','rating','timestamp'])  
# print(ratings.shape)  
# print(ratings.columns)  
users = pd.read\_csv('users.dat', sep='::', header=None, names=['user\_id', 'gender', 'age', 'occupation', 'zipcode'])  
# print(users.shape)  
# print(users.columns)  
# merging movie and ratings  
movie\_rating = pd.merge(movies, ratings, how='inner', on='movie\_id')  
# print(movie\_rating.shape)  
# merging movie\_rating and users  
final\_dataset = pd.merge(users, movie\_rating, how='inner', on='user\_id')  
print('Final Dataset Shape : ' + str(final\_dataset.shape))  
# Step 1 and 2 Completed  
# 3.1  
plt.figure(figsize=(10, 8))  
seaborn.countplot(x=final\_dataset['age'])  
plt.show()  
# analysis 1 : people between age 25 and 35 are more who watch movie  
# 3.2  
pd.value\_counts(final\_dataset.loc[final\_dataset['movie\_name'] == 'Toy Story (1995)', 'rating']).sort\_index().plot(kind='bar', title='Rating for Toy Story')  
plt.show()  
print('Toy Story is rated 4 many times')  
# 3.3  
aggregation = [np.mean, np.median, np.min, np.max]  
print('Top 25 movies according to the rating')  
datset = pd.DataFrame(final\_dataset.groupby('movie\_name').mean()[['rating']])  
print(datset.sort\_values('rating', ascending=False).head(25))  
# 3.4  
print('movies rated by the user 2696')  
print(final\_dataset.loc[final\_dataset['user\_id'] == 2696, ['user\_id', 'movie\_name', 'rating']])  
# 4.1  
gen\_dat = pd.DataFrame(final\_dataset.groupby('genere').mean()[['rating', 'age']])  
gen\_dat.sort\_values(by=['rating','age'], ascending=False)  
print('most rated movies types are Animation|Comedy|Thriller and Sci-Fi|War and rating is 4.5')  
# 4.2  
# get\_dummies create one hot encoding for genere  
hot\_end = pd.get\_dummies(final\_dataset['genere'])  
new\_data\_set = pd.concat([final\_dataset,hot\_end], axis=1)  
# 4.3  
# as we can notice all the column here is categorical column so visualizing all the plot  
cate = ['gender','occupation','genere','age']  
  
for i in cate:  
 table = pd.value\_counts(final\_dataset[i])  
 plt.style.use('dark\_background')  
 plt.pie(x=table, labels=table.index)  
 #seaborn.violinplot(x=table)  
 plt.show()  
  
print('plotting categorical values')  
  
for i in cate:  
 seaborn.countplot(x=i, hue='rating', data=final\_dataset)  
 plt.show()  
print('After visualizing all the variables the imp factor for rating is age and gender')  
  
plt.figure(figsize=(10,10))  
plt.subplot(1,2,1)  
seaborn.countplot(x='age', hue='rating', data=final\_dataset)  
plt.subplot(1,2,2)  
seaborn.countplot(x='gender', hue='rating', data=final\_dataset)  
plt.show()  
# 5  
# building the model  
#print(final\_dataset.columns)  
# removing unwanted columns  
final\_dataset.drop(['user\_id','zipcode','movie\_id','movie\_name','timestamp'], axis=1, inplace=True)  
final\_dataset['gender'] = final\_dataset['gender'].replace({'M': 0, 'F': 1})  
final\_dataset.drop(['genere'], axis=1, inplace=True)  
#new\_data = pd.concat([final\_dataset,hot\_end], axis=1)  
#new\_data.reset\_index()  
#new\_data = pd.concat([final\_dataset,hot\_end], axis=1)  
#new\_dataset = pd.DataFrame(new\_data)  
Y = pd.DataFrame(final\_dataset['rating'])  
final\_dataset.drop(['rating'], axis=1, inplace=True)  
#new\_dataset.drop(['rating'], axis=1, inplace=True)  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(final\_dataset, Y, test\_size=0.3)  
print(x\_train.shape)  
print(x\_test.shape)  
print(y\_train.shape)  
print(y\_test.shape)  
knn = KNeighborsClassifier(n\_neighbors=10)  
knn.fit(x\_train, y\_train)  
y\_pred = knn.predict(x\_test)  
print(classification\_report(y\_test, y\_pred))  
y\_pred = knn.predict(x\_test)