```
#Importing neccessary files
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```
# When working with Google colab and Google drive
from google.colab import drive
drive.mount('/content/drive')
%cd '/content/drive/MyDrive/Project2'
```

```
     Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
     /content/drive/MyDrive/Project2
```

```
#importing data
movies_Data = pd.read_csv("movies.dat",delimiter="::",names=["MovieID","Title","Genres"])
users_Data = pd.read_csv("users.dat",delimiter="::",names=["UserID","Gender","Age","Occupatio
ratings_Data = pd.read_csv("ratings.dat",delimiter="::",names=["UserID","MovieID","Rating","T
```

```
     /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: ParserWarning: Falling b

     /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: ParserWarning: Falling b
       This is separate from the ipykernel package so we can avoid doing imports until
     /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: ParserWarning: Falling b
       after removing the cwd from sys.path.
```

```
#printing the shape of the df
print(movies_Data.shape)
print(users_Data.shape)
print(ratings_Data.shape)
```

```
     (3883, 3)
     (6040, 5)
     (1000209, 4)
```

```
#1.Create a new dataset [Master_Data] with the following columns MovieID Title UserID Age Gen
masters_columns = ["MovieID","Title","UserID","Age","Gender","Occupation","Rating"]
masters_Data = pd.merge((pd.merge(movies_Data,ratings_Data,on="MovieID")),users_Data,on="User

masters_Data.head()
```

	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

```
#checking the datatype of the columns
masters_Data.dtypes
```

```
     MovieID        int64
     Title         object
     UserID         int64
     Age            int64
     Gender        object
     Occupation     int64
     Rating         int64
     dtype: object
```

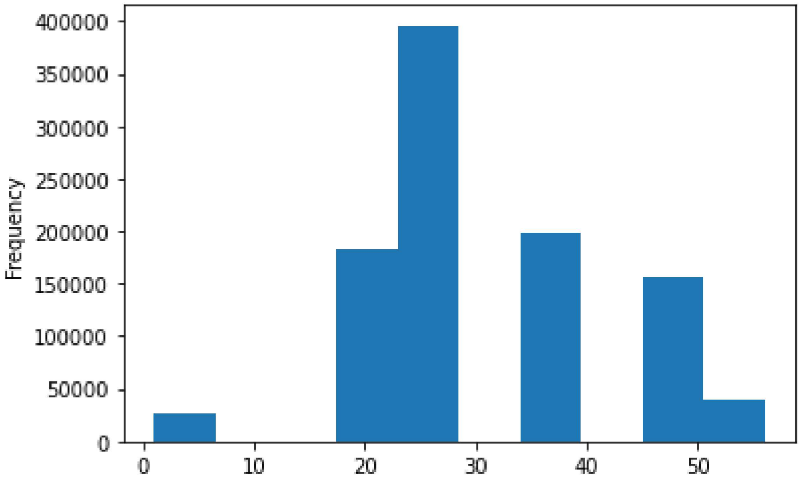
```
#printing the shape of the final df
masters_Data.shape
```

```
     (1000209, 7)
```

```
#Data Preprocessing

#User age grouping dictionary as per the data given
userAgeGroupDict={
 1:'Under 18',
 18:'18-24',
 25:'25-34',
 35:'35-44',
 45:'45-49',
 50:'50-55',
 56:'56+'
}

def userAgeGrp(age):
     if age<18:
             return userAgeGroupDict[1]
     elif age>=18 and age <=24:
             return userAgeGroupDict[18]
     elif age>=25 and age <=34:
             return userAgeGroupDict[25]
     elif age>=35 and age <=44:
             return userAgeGroupDict[35]
     elif age>=45 and age <=49:
             return userAgeGroupDict[45]
     elif age>=50 and age <=55:
         return userAgeGroupDict[50]
     elif age>=56 :
         return userAgeGroupDict[56]
```

```
#Adding age group to the User Data
users_Data["Age Group"] =  [userAgeGrp(age) for age in users_Data['Age']]



#Title and Year of release split into two columns so that year can be used to build model as
#defining a function to extract title and year
def extractTitle_Year(title):
 year = title[title.rfind('(')+1:title.rfind(')')]
 movieTitle = title[0:title.rfind('(')-1]
 return movieTitle, year

#assigning null columns for title and year
movies_Data['MovieTitle']=''
movies_Data['Year of Release']=''


for index, movie in movies_Data.iterrows():
     movieTitle,year = extractTitle_Year(movie.Title)
     movies_Data.at[index,'MovieTitle']=movieTitle
     movies_Data.at[index,'Year of Release']=year


#2.Explore the datasets using visual representations (graphs or tables), also include your co
# a) User Age Distribution
masters_Data.Age.plot(kind='hist')
#plotting a histogram to visually represent the age of all the users.
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4d8e176510>
```



```
# b) User rating of the movie “Toy Story”
ToyStory= masters_Data.loc[masters_Data['Title'].str.contains("Toy Story",case=False)]
with pd.option_context('display.max_rows', None, 'display.max_columns', None):
    display(ToyStory)
#checking whether the Title contains the words Toy story irrespective of the case and assigni
```

480522	1	Toy Story (1995)	4975	35	M	0	3
480697	1	Toy Story (1995)	4979	35	M	2	5
481579	1	Toy Story (1995)	4983	45	F	16	4
481703	3114	Toy Story 2 (1999)	4983	45	F	16	5
481722	1	Toy Story (1995)	4989	25	M	0	5
481786	3114	Toy Story 2 (1999)	4989	25	M	0	5
481799	1	Toy Story (1995)	4990	25	F	4	3
481855	3114	Toy Story 2 (1999)	4990	25	F	4	3
481860	1	Toy Story (1995)	4995	50	M	20	4
482216	1	Toy Story (1995)	4998	18	F	4	3
482295	1	Toy Story (1995)	4999	56	F	13	5
482351	3114	Toy Story 2 (1999)	4999	56	F	13	5
482363	1	Toy Story (1995)	5005	45	M	16	4
482792	1	Toy Story (1995)	5009	45	F	16	3
482855	1	Toy Story (1995)	5011	18	M	4	4
483414	1	Toy Story (1995)	5015	35	M	6	5
483942	3114	Toy Story 2 (1999)	5015	35	M	6	5
484058	1	Toy Story (1995)	5026	25	M	17	4
484797	3114	Toy Story 2 (1999)	5026	25	M	17	3
484932	1	Toy Story (1995)	5032	18	M	17	4
485017	3114	Toy Story 2 (1999)	5032	18	M	17	5
485035	1	Toy Story (1995)	5034	25	F	15	4
485176	1	Toy Story (1995)	5035	25	F	17	5
485392	3114	Toy Story 2 (1999)	5035	25	F	17	5
485425	1	Toy Story (1995)	5037	35	M	12	5
485517	3114	Toy Story 2 (1999)	5037	35	M	12	5
485530	1	Toy Story (1995)	5038	25	M	20	5
485684	3114	Toy Story 2 (1999)	5038	25	M	20	4
485740	1	Toy Story (1995)	5042	18	F	2	4
486066	3114	Toy Story 2 (1999)	5042	18	F	2	4
486108	1	Toy Story (1995)	5046	25	M	16	5
486821	3114	Toy Story 2 (1999)	5046	25	M	16	5