```
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
import csv
```

```
_user = pd.read_csv("u.user", engine='python', sep='|', names=["userID", "Age",
```

```
df_user.head()
```

	userID	Age	Gender	occupation	Zip-Code
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213

```
df_movie = pd.read_csv("u.item", engine='python', sep='|', names=["movieID", "Mo
             "IMDb URL", "unknown", "Action", "Adventure", "Animation",
             "Children's", "Comedy", "Crime", "Documentary", "Drama", "Fantasy"
             "Film-Noir", "Horror", "Musical", "Mystery", "Romance", "Sci-Fi",
             "Thriller", "War", "Western"])
```

```
df_movie.head()
```

	movieID	Movie title	release date	video release date	IMDb URL	unknown	Action
0	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0
1	2	GoldenEye (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?GoldenEye%20(...	0	1
2	3	Four Rooms (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Four%20Rooms%...	0	0
3	4	Get Shorty (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Get%20Shorty%...	0	1
4	5	Copycat (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Copycat%20(1995)	0	0

```
df_ratings = pd.read_csv("u.data", engine='python', sep='\t', names=["userID", "
```

```
df_ratings.head()
```

	userID	movieID	Rating	Timestamp
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596

```
df_merged1 = df_movie.merge(df_ratings, how='outer')
```

```
df_merged1.head()
```

	movieID	Movie title	release date	video release date	IMDb URL	unknown	Action	A
0	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	
1	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	
2	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	
3	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	
4	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	

```
#Merging Users and Ratings
df_merged2 = df_user.merge(df_ratings, how='inner')
```

```
df_merged2.head()
```

	userID	Age	Gender	occupation	Zip-Code	movieID	Rating	Timestamp
0	1	24	M	technician	85711	61	4	878542420
1	1	24	M	technician	85711	189	3	888732928
2	1	24	M	technician	85711	33	4	878542699
3	1	24	M	technician	85711	160	4	875072547
4	1	24	M	technician	85711	20	4	887431883

```
#Merging Users/Ratings/Movies
df_merged3 = df_merged1.merge(df_merged2, how='inner')
```

```
df_merged3.head()
```

	movieID	Movie title	release date	video release date	IMDb URL	unknown	Action	A
0	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	
1	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	
2	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	
3	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	
4	1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	

```
df_merged3 = df_merged3.fillna(0)
df_merged3.UserID = df_merged3.userID.astype(int)
df_merged3.Rating = df_merged3.Rating.astype(int)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarnin
```

```
df_merged3.shape
```

```
(100000, 31)
```

```
df_merged3.sort_values(by=['userID'], ascending=True)
```

	movieID	Movie title	release date	video release date	IMDb URL	unknown	Act
5529	39	Strange Days (1995)	01-Jan-1995	0.0	http://us.imdb.com/M/title-exact?Strange%20Day...	0	
11183	82	Jurassic Park (1993)	01-Jan-1993	0.0	http://us.imdb.com/M/title-exact?Jurassic%20Pa...	0	
1915	10	Richard III (1995)	22-Jan-1996	0.0	http://us.imdb.com/M/title-exact?Richard%20III...	0	
23425	168	Monty Python and the Holy Grail (1974)	01-Jan-1974	0.0	http://us.imdb.com/M/title-exact?Monty%20Pytho...	0	
6072	47	Ed Wood (1994)	01-Jan-1994	0.0	http://us.imdb.com/M/title-exact?Ed%20Wood%20(...	0	
...	...	...	...	...	...	...	
2159	11	Seven (Se7en) (1995)	01-Jan-1995	0.0	http://us.imdb.com/M/title-exact?Se7en%20(1995)	0	
57812	415	Apple Dumpling Gang, The (1975)	01-Jan-1975	0.0	http://us.imdb.com/M/title-exact?Apple%20Dumpl...	0	
30525	201	Evil Dead II (1987)	01-Jan-1987	0.0	http://us.imdb.com/M/title-exact?Evil%20Dead%2...	0	
54425	373	Judge Dredd (1995)	01-Jan-1995	0.0	http://us.imdb.com/M/title-exact?Judge%20Dredd...	0	
81113	722	Nine Months (1995)	01-Jan-1995	0.0	http://us.imdb.com/M/title-exact?Nine%20Months...	0	

100000 rows × 31 columns

```
#Rearranging merged3 columns into suitable format
master_data = df_merged3[['userID', 'movieID', 'Movie title', 'Rating', 'unknown
```

```
master_data.head()
```

	userID	movieID	Movie title	Rating	unknown	Action	Adventure	Animation	Chi
0	308	1	Toy Story (1995)	4	0	0	0	1	
1	287	1	Toy Story (1995)	5	0	0	0	1	
2	148	1	Toy Story (1995)	4	0	0	0	1	
3	280	1	Toy Story (1995)	4	0	0	0	1	
4	66	1	Toy Story (1995)	3	0	0	0	1	

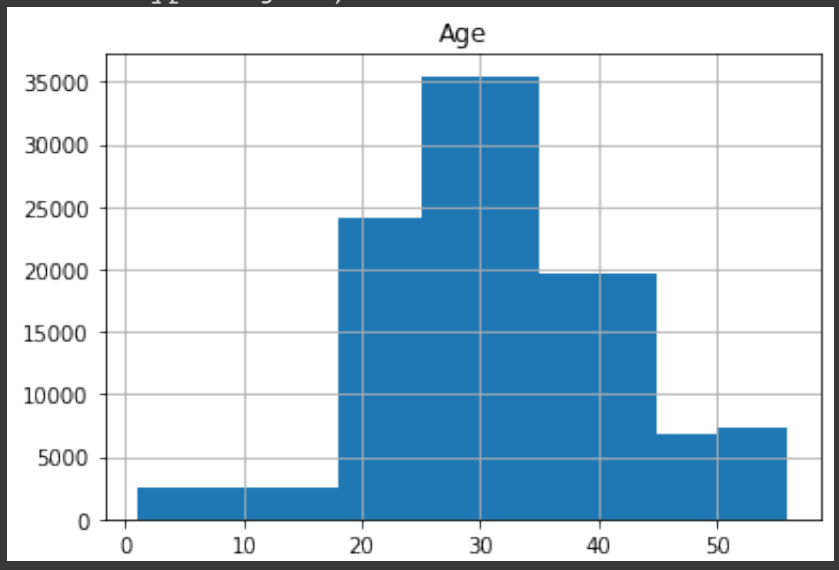
```
master_data.sort_values(by=['userID'], ascending=True)
```

	userID	movieID	Movie title	Rating	unknown	Action	Adventure	Animatic
5529	1	39	Strange Days (1995)	4	0	1	0	
11183	1	82	Jurassic Park (1993)	5	0	1	1	
1915	1	10	Richard III (1995)	3	0	0	0	
23425	1	168	Monty Python and the Holy Grail (1974)	5	0	0	0	
6072	1	47	Ed Wood (1994)	4	0	0	0	
...	...	...	...	...	...	...	...	
2159	943	11	Seven (Se7en) (1995)	4	0	0	0	
57812	943	415	Apple Dumpling Gang, The (1975)	1	0	0	0	
30525	943	201	Evil Dead II (1987)	5	0	1	1	
54425	943	373	Judge Dredd (1995)	3	0	1	1	
81113	943	722	Nine Months (1995)	3	0	0	0	

100000 rows × 28 columns

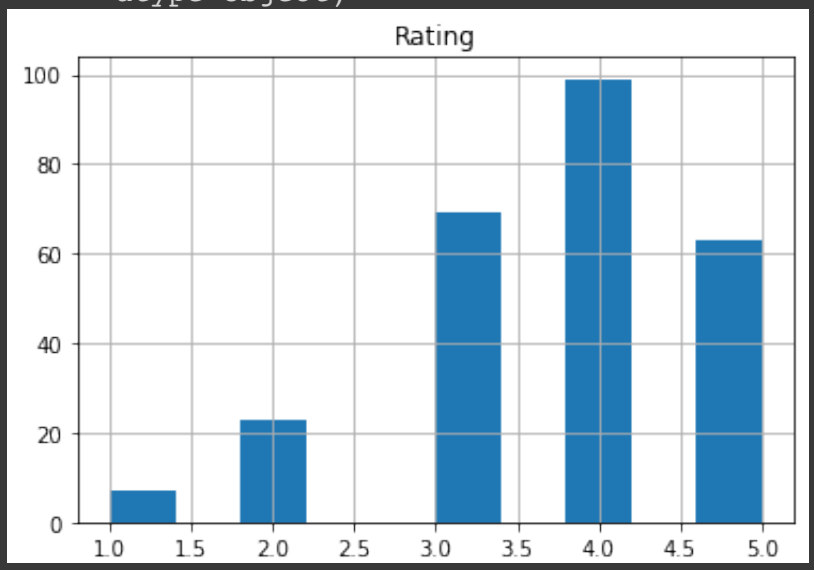
```
#PERFORMING EDAs
bins_list = [1, 18, 25, 35, 45, 50, 56]
master_data.hist(column='Age', bins = bins_list)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fc294bd8110>]],
      dtype=object)
```




```
#Checking ratings on Jurassic Park
master_data[master_data['movieID'] == 82].hist(column='Rating')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fc28f75cd90>]],
      dtype=object)
```




```
#Avg rating of Jurassic park
master_data[master_data['movieID'] == 82].Rating.mean()
```

```
3.7203065134099615
```

```
#Predictive analysis
#Analysis factors affecting movie rating
master_data.head()
master_data['Gender'].replace(['F','M'],[0,1],inplace=True)
md_small = master_data.iloc[:, [1, 2, 3, 23, 24, 25, 26]]
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/series.py:4582: SettingW
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
  method=method,
```

```
md_small.head()
```

	movieID	Movie title	Rating	Zip-Code	Gender	Age	occupation
0	1	Toy Story (1995)	4	95076	1	60	retired
1	1	Toy Story (1995)	5	31211	1	21	salesman
2	1	Toy Story (1995)	4	97006	1	33	engineer
3	1	Toy Story (1995)	4	22903	0	30	librarian
4	1	Toy Story (1995)	3	80521	1	23	student

```
#Convert as many dtypes into int to get better coef insights
md_small.dtypes
```

```
movieID         int64
Movie title    object
Rating          int64
Zip-Code       object
Gender          int64
Age             int64
occupation     object
dtype: object
```

```
#Finding coorelation coef
md_small[md_small.columns[1:]].corr()['Rating'][:]
```

```
Rating    1.000000
Gender   -0.000862
Age       0.054460
Name: Rating, dtype: float64
```

```
temp_genre = master_data.iloc[:, [4, 5, 6, 7, 8, 9 , 10, 11, 12, 13, 14, 15, 16,
```

```
temp_genre.head()
```

	unknown	Action	Adventure	Animation	Children's	Comedy	Crime	Documen
0	0	0	0	1	1	1	0	
1	0	0	0	1	1	1	0	
2	0	0	0	1	1	1	0	
3	0	0	0	1	1	1	0	
4	0	0	0	1	1	1	0	

```
master_features = pd.merge(md_small, temp_genre, left_index=True, right_index=Tr
```

```
master_features.head()
```

	movieID	Movie title	Rating	Zip-Code	Gender	Age	occupation	unknown	Action	Ac
0	1	Toy Story (1995)	4	95076	1	60	retired	0	0	
1	1	Toy Story (1995)	5	31211	1	21	salesman	0	0	
2	1	Toy Story (1995)	4	97006	1	33	engineer	0	0	
3	1	Toy Story (1995)	4	22903	0	30	librarian	0	0	
4	1	Toy Story (1995)	3	80521	1	23	student	0	0	

```
master_features.dtypes
```

```
movieID         int64
Movie title    object
Rating          int64
Zip-Code       object
Gender          int64
Age             int64
occupation     object
unknown         int64
Action          int64
Adventure       int64
Animation       int64
Children's      int64
Comedy          int64
Crime           int64
Documentary     int64
Drama           int64
Fantasy         int64
Film-Noir       int64
Horror          int64
Musical         int64
Mystery         int64
Romance         int64
Sci-Fi          int64
Thriller        int64
War             int64
Western         int64
dtype: object
```

```
#Preparing data for linear regression
#Drop all obj dtype
X_feature = md_small.drop(['Zip-Code', 'Movie title', 'occupation'], axis=1)
```

```
X_feature.head()
```

	movieID	Rating	Gender	Age
0	1	4	1	60
1	1	5	1	21
2	1	4	1	33
3	1	4	0	30
4	1	3	1	23

```
#Preparing to train on first 40 movies
X_feature_small = X_feature[X_feature['movieID'] < 40]
```

```
X_feature_small_trimmed = X_feature_small.drop(['movieID','Rating'], axis=1)
X_feature_small_trimmed.head()
```

	Gender	Age
0	1	60
1	1	21
2	1	33
3	0	30
4	1	23

```
Y_target = master_features['Rating'][master_features['movieID']< 40]
```

```
x_train, x_test, y_train, y_test = train_test_split(X_feature_small_trimmed,Y_ta
```

```
logreg = LogisticRegression(max_iter=100000)
```

```
logreg.fit(x_train,y_train)
```

```
LogisticRegression(max_iter=100000)
```

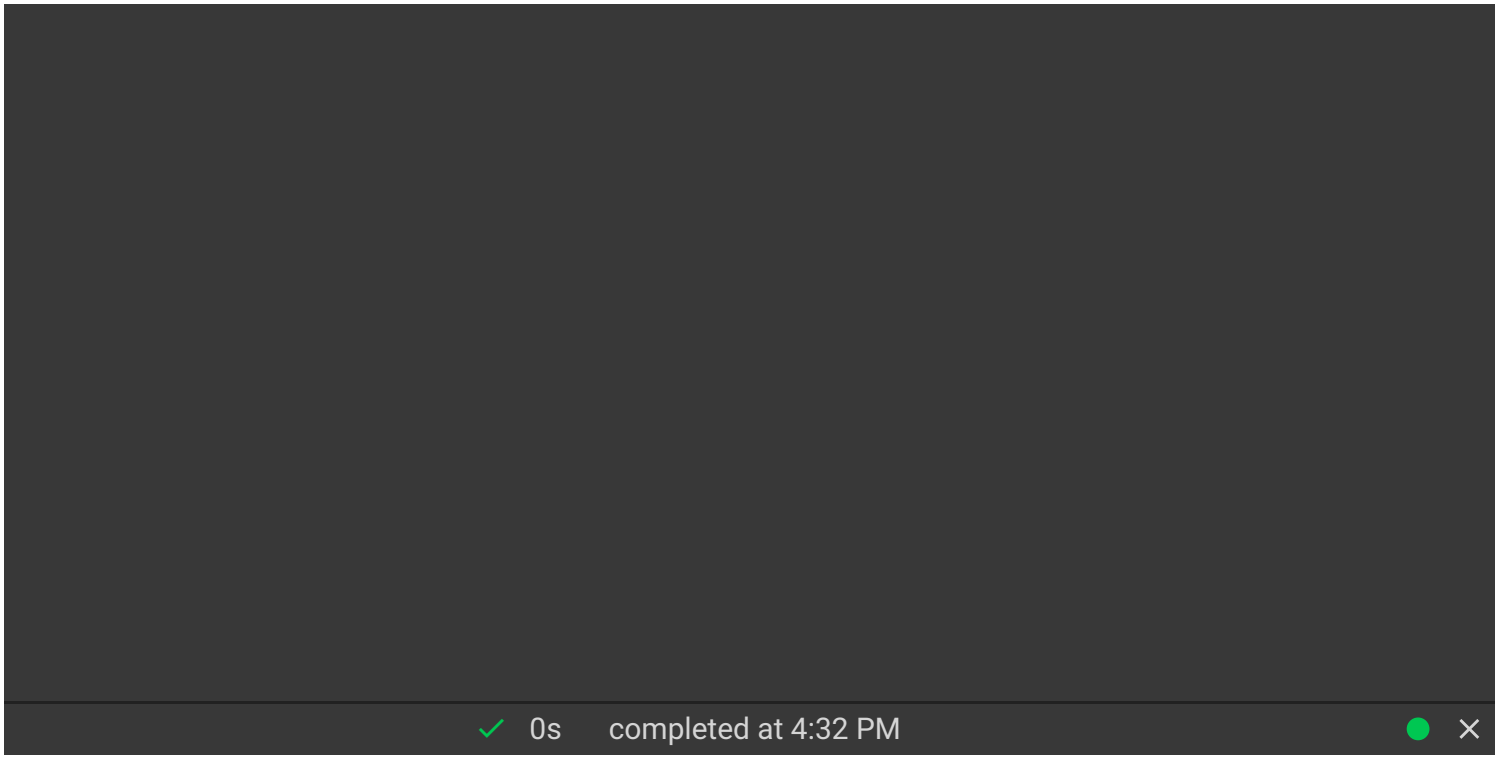
```
y_pred = logreg.predict(x_test)
```

```
metrics.accuracy_score(y_test,y_pred)
```

```
0.3533190578158458
```

```
print ('actual:    ', y_test.values[0:30])
print ('predicted: ', y_pred[0:30])
```

```
actual:     [5 2 4 5 5 5 3 4 4 4 2 4 5 3 5 4 1 3 3 1 4 4 4 4 2 5 5 4 4 3]
predicted:  [4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4]
```


0s completed at 4:32 PM