"Classification on Movies"

Guided by-Mrs.Shubhagi Kale(course master)
Name of the guide- Shubhagi kale

Presented By-

Snehal Bankar

206

Ajay Bhapkar

207

Sayali Deshmukh

214

Introduction

- A movie dataset is a structured collection of information about movies that is used for various purposes, such as analysis, research, and machine learning.
- It typically contains a wide range of data points related to movies, including details about the film's title, release year, genre, director, cast, plot summary, ratings, and other relevant information.
- Movie datasets are often curated from various sources, including movie databases, film industry websites, user-generated reviews, and online platforms
- These datasets are powerful tools for studying the cultural impact of movies

Motivation

- **Film Industry Analysis: Movie datasets allow researchers and analysts to gain insights into the film industry, such as box office performance, revenue trends, production budgets, and the success factors associated with movies. So we take the motivation that Movie datasets provide valuable information about audience preferences, allowing researchers to understand the factors that drive movie choices, genre preferences, and viewing habits.**

Details of Dataset

Name :- Movies Dataset

Number of Features :-

Movie title : The name or title of the movie.
Release year : The year in which the movie was released.
Genre : The category or genre of the movie (e.g., action, comedy,drama)
Director : The name of the director who helmed the movie

Number of records :- Rows:5043
Columns:28

Data Manipulation

Pandas offers efficient data structures like DataFrames and Series, which allow for flexible and intuitive data manipulation. It provides a wide range of functions and methods for tasks such as filtering, selecting, transforming, and aggregating data. With pandas, data scientists can easily clean, preprocess, and reshape data to suit their analysis needs.

```
#df1=pd.read_csv("/content/MOVIES DATASET1.csv",usecols=
['director_facebook_likes'])
print(df1.max())
```

output: director_facebook_likes

0	0.0
1	563.0
2	0.0
3	22000.0
4	131.0
...	...
5038	2.0
5039	NaN
5040	0.0
5041	0.0
5042	16.0

```
# find number of movies released in USA in the year 2000
df2 = df.groupby(['country','title_year']).get_group(('USA',2000)).count()
print(df2['movie_title'])
```

output: 136

```
# count the number of black and white movies

df3 = df.groupby('color').get_group(' Black and White').count()
print(df3['movie_title'])
```

output:209

```
# convert name of actor_2 column in lowercase
print(df['actor_2_name'].str.lower())
```

```
output:    joel david moore
1          orlando bloom
2           rory kinnear
3         christian bale
4            rob walker
                ...
5038       daphne zuniga
5039       valorie curry
5040      maxwell moody
5041       daniel henney
5042    brian herzlinger
```

```
# count the number of movies realeased in perticular country (count the
                accurrences of each unique country in a column)
                print(df['country'].value_counts())
```

output:

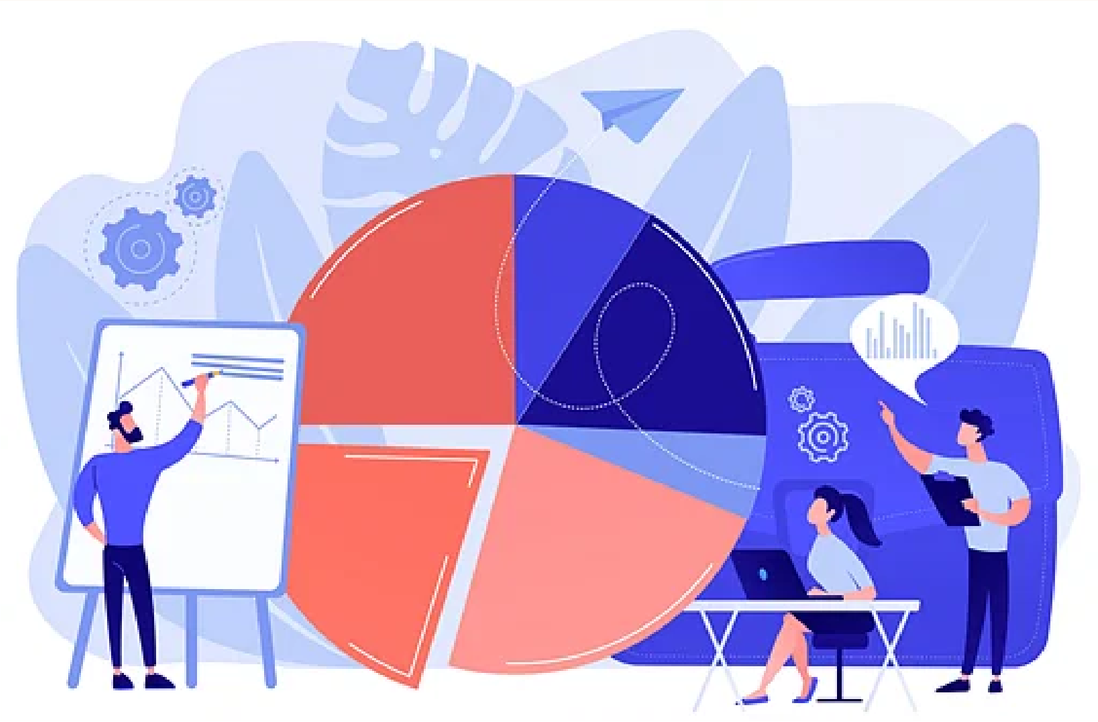
USA	2987	South Korea	8	Indonesia	1
UK	318	Denmark	8	Israel	1
France	101	Ireland	7	Poland	1
Germany	80	Mexico	6	Colombia	1
Canada	59	Brazil	5	New Line	1
Australia	39	India	5	Iceland	1
Spain	21	Iran	4	Aruba	1
Japan	15	Thailand	4	Peru	1
Hong Kong	13	Norway	4	Belgium	1
China	13	Russia	3	Georgia	1
Italy	11	Argentina	3	West Germany	1

Data Visualization

Data visualization is a field in data analysis that deals with visual representation of data. It graphically plots data and is an effective way to communicate inferences from data.

Using data visualization, we can get a visual summary of our data. With pictures, maps and graphs, the human mind has an easier time processing and understanding any given data. Data visualization plays a significant role in the representation of both small and large data sets, but it is especially useful when we have large data sets, in which it is impossible to see all of our data, let alone process and understand it manually.

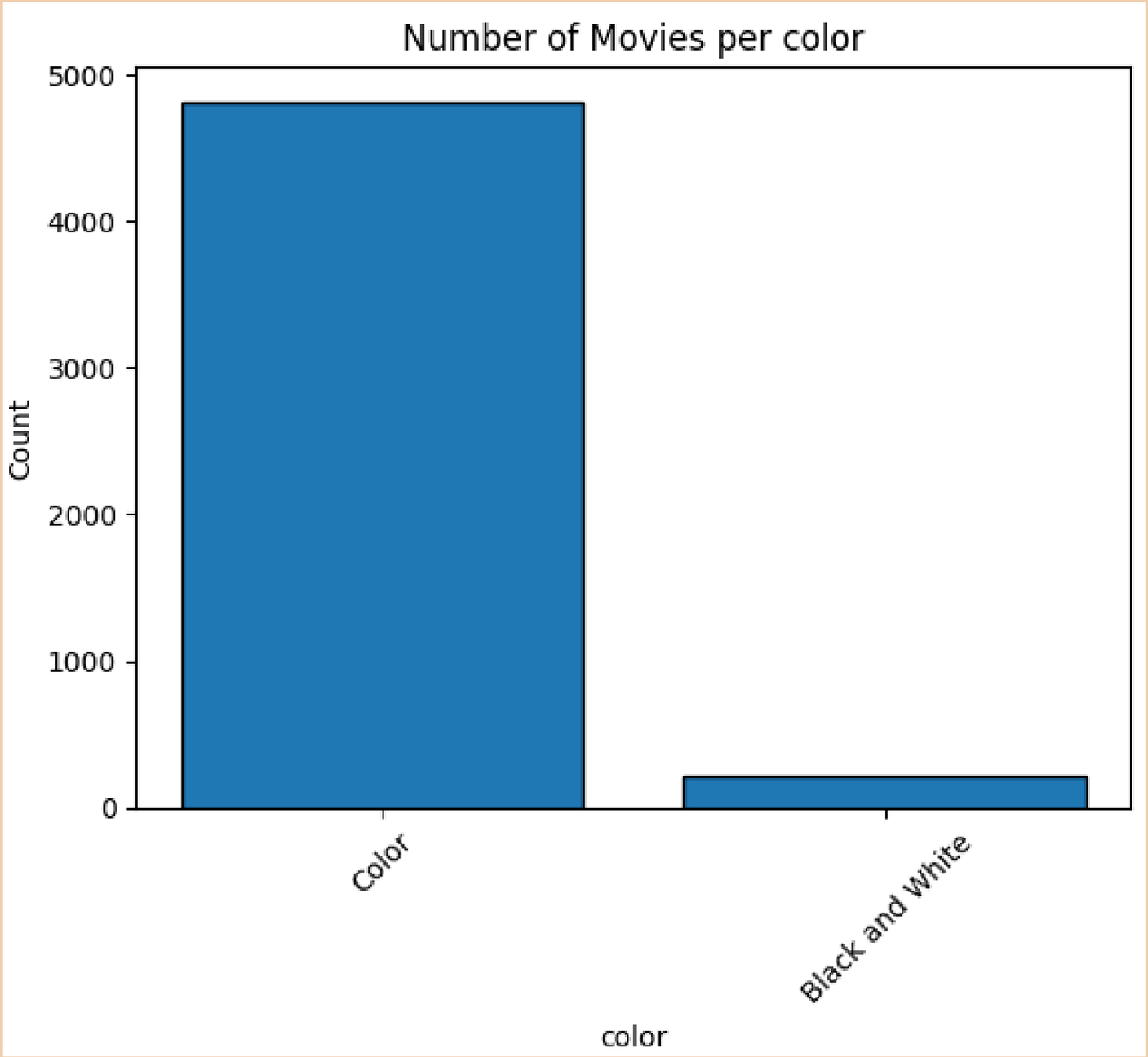
Data Visualization in Python



Python offers several plotting libraries, namely Matplotlib, Seaborn and many other such data visualization packages with different features for creating informative, customized, and appealing plots to present data in the most simple and effective way.

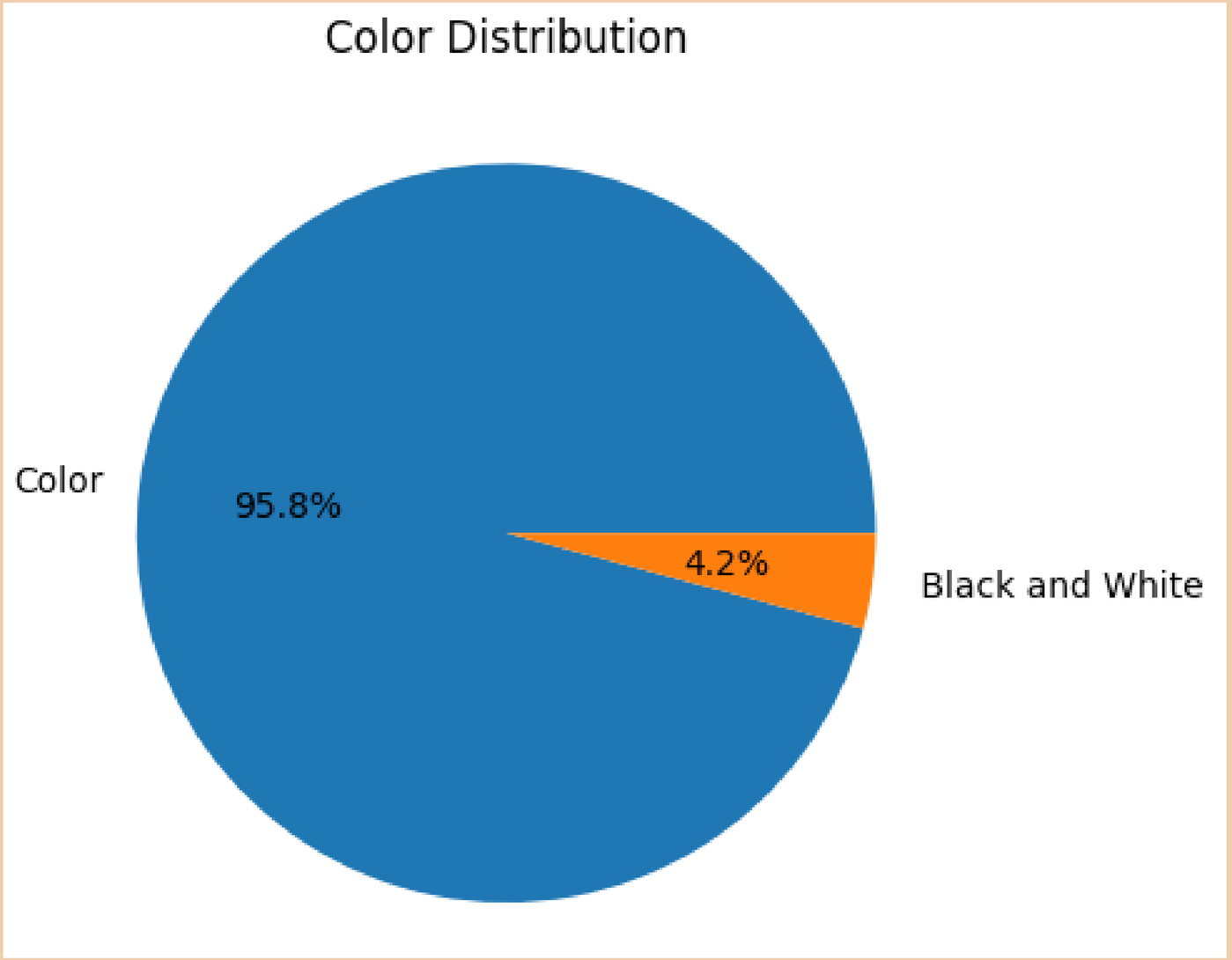
```
import matplotlib.pyplot as plt
import pandas as pd
movies = pd.read_csv('MOVIES DATASET.csv')
color_counts = movies['color'].value_counts()
plt.bar(color_counts.index, color_counts.values, edgecolor='black')
plt.xlabel('color')
plt.ylabel('Count')
plt.title('Number of Movies per color')
plt.xticks(rotation=45)
plt.show()
```

output:


Number of Movies per color
5000
4000
3000
2000
1000
0
Count
Color
Black and White
color

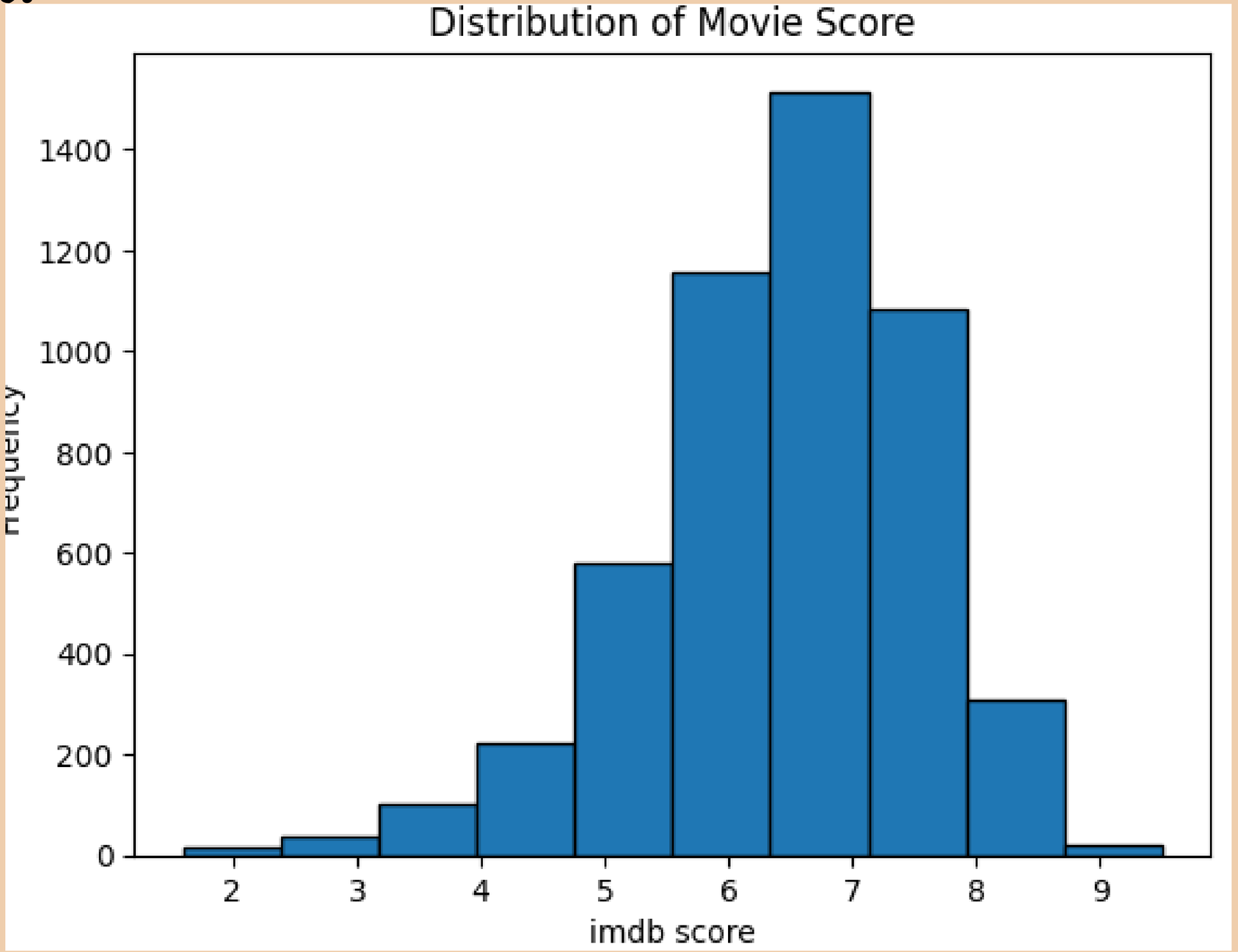
```
import matplotlib.pyplot as plt
import pandas as pd
movies = pd.read_csv('MOVIES DATASET.csv')
# Assuming you have a DataFrame called
'movies' with a column 'color'
color_counts = movies['color'].value_counts()
plt.pie(color_counts.values,
labels=color_counts.index, autopct='%1.1f%%')
plt.title('Color Distribution')
plt.show()
```

output:



```
import matplotlib.pyplot as plt
import pandas as pd

movies = pd.read_csv('MOVIES DATASET.csv')

# Assuming you have a DataFrame called 'movies' with a column 'imdb_score'
plt.hist(movies['imdb_score'], bins=10, edgecolor='black')
plt.xlabel('imdb score')
plt.ylabel('Frequency')
plt.title('Distribution of Movie Score')
plt.show()
```

output:


Distribution of Movie Score
1400
1200
1000
800
600
400
200
0
2
3
4
5
6
7
8
9
imdb score

Predictive Technique (LR/KNN/KMeans)

1. A linear regression is a statistical model that attempts to show the relationship between two variables with a linear equation.

2. A regression analysis involves graphing a line over a set of data points that most closely fits the overall shape of the data.

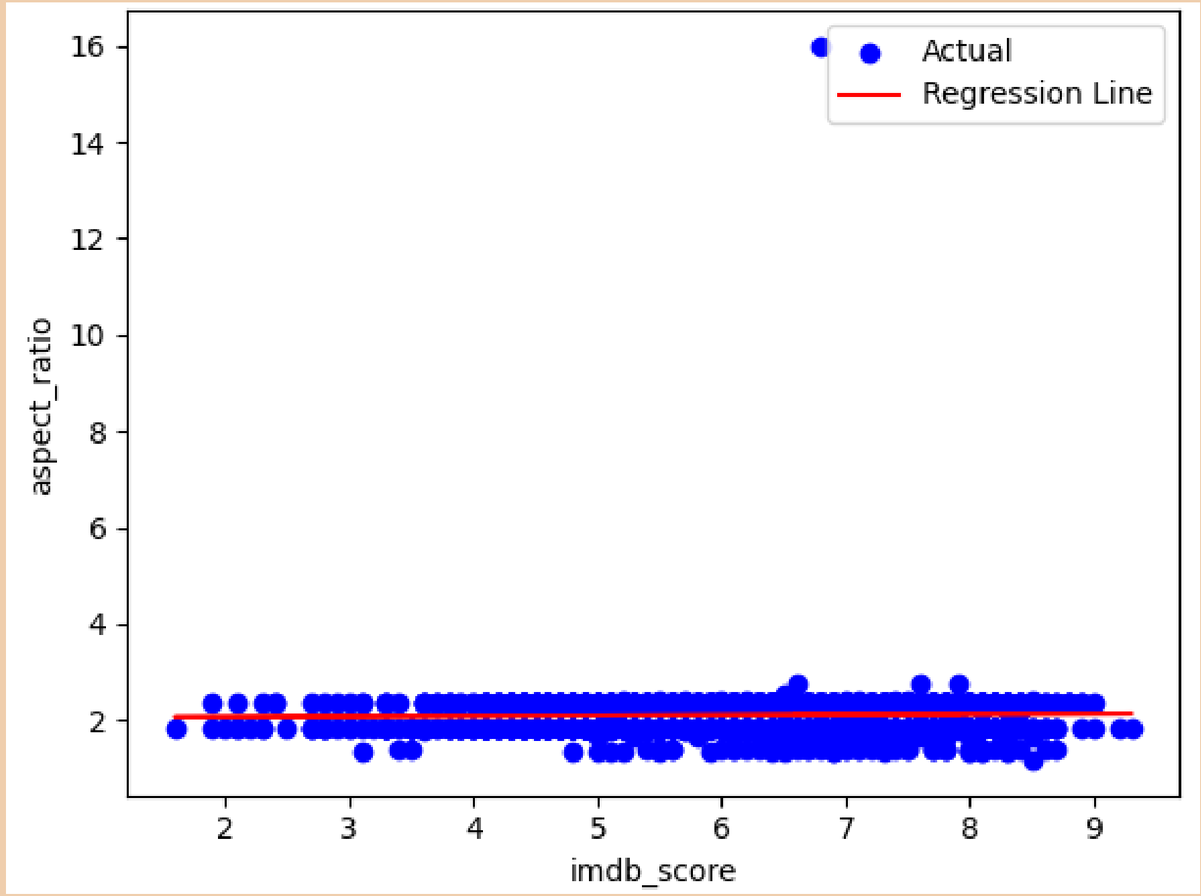
3. A regression shows the extent to which changes in a "dependent variable," which is put on the y-axis, can be attributed to changes in an "explanatory variable," which is placed on the x-axis.

Linear Regression

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns from sklearn.linear_model import LinearRegression
df1=pd.read_csv("/content/sample_data/MOVIES DATASET1.csv") data =
df1.dropna() print(data) # Extract the columns for linear regression X =
data['imdb_score'].values.reshape(-1, 1) # Input feature y =
data['aspect_ratio'].values # Target variable # Create and fit the linear
regression model model = LinearRegression() model.fit(X, y) # Predict the
target variable y_pred = model.predict(X) # Plot the data points and the
regression line plt.scatter(X, y, color='blue', label='Actual') plt.plot(X,
y_pred, color='red', label='Regression Line') plt.xlabel('imdb_score')
plt.ylabel('aspect_ratio') plt.legend() plt.show()
```

output:

color director_name
num_critic_for_reviews duration \
0 Color James Cameron 723.0 178.0
1 Color Gore Verbinski 302.0 169.0
2 Color Sam Mendes 602.0 148.0
3 Color Christopher Nolan 813.0 164.0
5 Color Andrew Stanton 462.0 132.0
... ...
5026 Color Olivier Assayas 81.0 110.0
5027 Color Jafar Panahi 64.0 90.0
5033 Color Shane Carruth 143.0 77.0
5035 Color Robert Rodriguez 56.0 81.0
5042 Color Jon Gunn 43.0 90.0



KNN

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.axes as ax
from sklearn.metrics import classification_report,\
confusion_matrix
df = pd.read_csv('/content/sample_data/MOVIES DATASET1.csv')
df = df.dropna()
X=df['num_critic_for_reviews']
df=df.dropna()
Y=df['duration']
X=np.array(df['num_critic_for_reviews']).reshape(-1,1)
Y=np.array(df['duration']).reshape(-1,1)
X_train, X_test,y_train, y_test = train_test_split(X,Y,test_size=0.30)
from sklearn.metrics import classification_report,\
confusion_matrix
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train, y_train)
pred = knn.predict(X_test)
print(confusion_matrix(y_test, pred))
print(classification_report(y_test, pred))
```

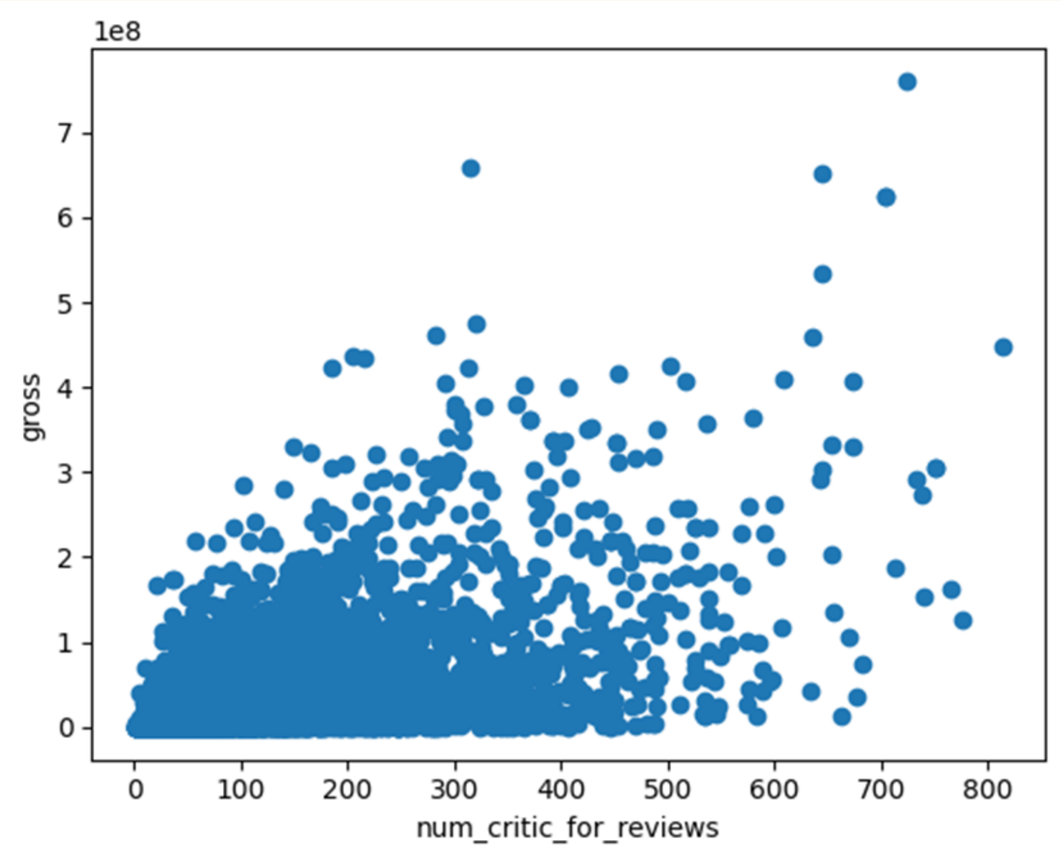
output:

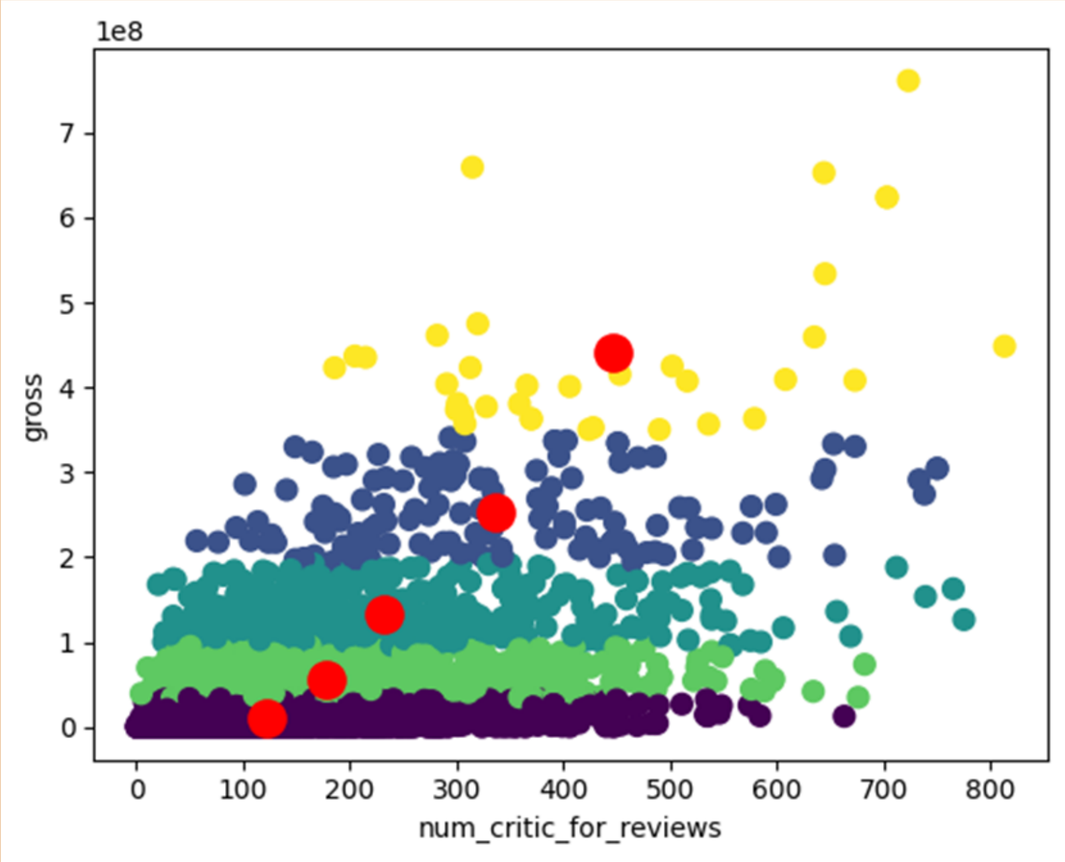
```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

	precision	recall	f1-score	support
45.0	0.00	0.00	0.00	0
63.0	0.00	0.00	0.00	1
66.0	0.00	0.00	0.00	1
69.0	0.00	0.00	0.00	1
72.0	0.00	0.00	0.00	1
74.0	0.00	0.00	0.00	1
75.0	0.00	0.00	0.00	3
76.0	0.00	0.00	0.00	1
77.0	0.00	0.00	0.00	2

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

df = pd.read_csv("/content/sample_data/MOVIES DATASET1.csv")
Data = {'x': df["num_critic_for_reviews"], 'y': df["gross"]}
df=pd.DataFrame(Data, columns=['x', 'y'])

plt.xlabel("num_critic_for_reviews")

plt.ylabel("gross")

plt.scatter(df['x'], df['y'])
plt.show()
df.dropna(inplace=True)

km = KMeans(n_clusters=5).fit(df)
centroids = km.cluster_centers_

plt.xlabel("num_critic_for_reviews")
plt.ylabel("gross")
plt.scatter(df['x'], df['y'], c=km.labels_.astype(float), s=60, alpha=1)
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=190)
plt.show()
```

output:


1e8
7
6
5
4
3
2
1
0
gross
0
100
200
300
400
500
600
700
800
num_critic_for_reviews


1e8
7
6
5
4
3
2
1
0
gross
0
100
200
300
400
500
600
700
800
num_critic_for_reviews

Application

1. Filtering values on the basis of given condition

2. Apply a certain function to create either a new variable or perform related operations

3. Helps to visualize various results of data

4. It helps to understand all about movies, which movie is best, who was actors in that movie, what was budget and many more things

Conclusion

Data scientists today draw largely from extensions of the “analyst” of years past trained in traditional disciplines. As data science becomes an integral part of many industries and enriches research and development, there will be an increased demand for more holistic and more nuanced data science roles.

Thank you !