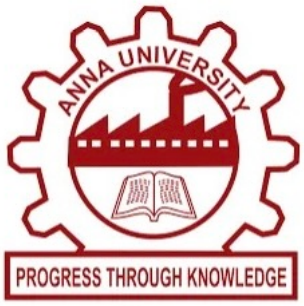
University College of Engineering Villupuram



Department of computer Science and Engineering

Experience Based Project Learning - IBM (E2324)
Conducted by IBM

Personalized content recommendation system
phase - 3 Document

Submitted by

1. S.Akalya
2. K.Harinitha
3. G.Muthazhagi
4. E.S.Mangala Yazhini
5. PC.Vaishanavi

Personalized content recommendation system
Phase 3 : Exploratory data analysis and Data Visualization

Introduction :

Data visualization is the process of using visual elements like charts, graphs, or maps to represent data. It translates complex, high-volume, or numerical data into a visual representation that is easier to process. It benefits them to recognize new patterns and errors in the data.To build a movie recommender system ,EDA helps in understanding the distribution of ratings, genres, user demographics, etc. This knowledge is essential for designing effective recommendation algorithms.

Objectives of Exploratory data analysis :

It involves analyzing and visualizing data to understand its key characteristics, uncover patterns, and identify relationships between variables refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables.

The primary objectives of EDA include :

1. **Understand Data Structure**: Identify the underlying structure and characteristics of the data. This involves understanding the types of variables (categorical, numerical), data types, and the relationships between variables.

2. **Detect Outliers and Anomalies**: Identify any unusual data points or anomalies that may require further investigation or correction. Outliers can significantly impact the results of data analysis and modeling.

3. **Uncover Patterns and Relationships**: Discover patterns, trends, and relationships between variables. This can involve examining correlations, associations, and dependencies.

Dataset Description :

The TMDb dataset used to build content based recommendation system .The TMDb (The Movie Database) is a comprehensive movie database that provides information about movies.The movie dataset csv file contains movie title, movie id ,release date, run time ,vote average , genres ,keywords , tagline ,overview of the movie etc . The credit dataset csv file contains cast and crew of movie . These datasets are used to build the content based movie recommendation system.

Data Visualization Techiniques :

Data visualization is the process of using visual elements like charts, graphs, or maps to represent data. It translates complex, high-volume, or numerical data into a visual representation that is easier to process.Data visualization is the process of creating graphical representations of information. This process helps the presenter communicate data in a way that's easy for the viewer to interpret and draw conclusions. Univariate, Bivariate and Multivariate Analysis are used to understand the dataset .

Import the libraries :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the dataset :

```
movies = pd.read_csv("Movies.csv")
credit = pd.read_csv("Credit.csv")
```

Merge the dataset :

The dataset is merged to build the recommendation system .

```
df = movies.merge(credit , on = "title")
```

Preprocessing the dataset for Exploratory data analysis :

The preprocessing step is performed to convert the datatypes of each column into suitable form for exploratory data analysis using the data preprocessing techniques used in phase 2 of personalized content recommendation system.

Applying Data visualization techniques after the columns are preprocessed into suitable columns :

1.Univariate Analysis :

Univariate Analysis is a type of data visualization where we visualize only a single variable at a time. Univariate graphs are used to visualize the distribution and frequency of a single variable.

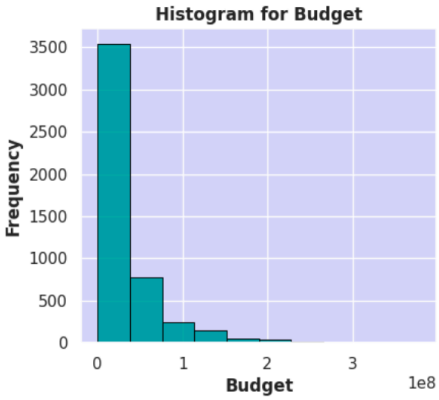
1. **Bar Graph :**
   Displays the frequency or proportion of different categories of a single variable.
   Example: A bar graph showing the number of students in different majors .
2. **Histogram :**
   Represents the frequency distribution of a continuous numerical variable by dividing the data into bins or intervals.
   Example: A histogram showing the distribution of exam scores for a class.
3. **Pie Chart :**
   Shows the relative proportions or percentages of different categories of a single variable as slices of a circle.
   Example: A pie chart displaying the market share of different smartphone brands.

Histogram for Budget :
code :

```
plt.figure(figsize = (4.5,4))
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white'})
sns.histplot(data = df , x = "budget" , bins = 10 , color = "darkcyan",
edgecolor = "black" ,linewidth = 0.65)
plt.xlabel("Budget",weight='bold')
```

```
plt.ylabel("Frequency",weight='bold')
plt.title("Histogram for Budget",weight='bold')
plt.show()
```

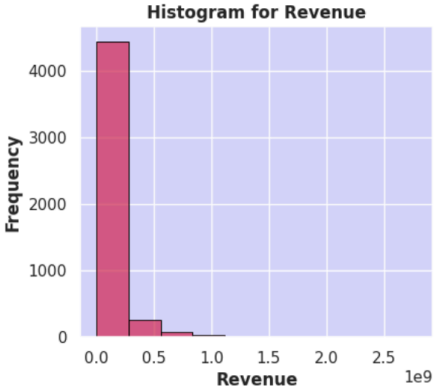
output :



Histogram for Revenue :
code :

```
plt.figure(figsize = (4.5,4))
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white'})
sns.histplot(data = df , x = "revenue" , bins = 10 ,color = "#c23d5e",
edgecolor = "black" ,linewidth = 0.65)
plt.xlabel("Revenue",weight='bold')
plt.ylabel("Frequency",weight='bold')
plt.title("Histogram for Revenue",weight='bold')
plt.show()
```

output :

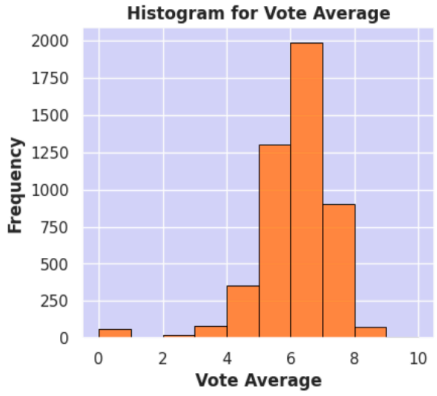


Histogram for Vote Average :

code :

```
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white'})
sns.histplot(data = df , x = "vote_average" , bins = 10 , color = "#ff751a",
edgecolor = "black" ,linewidth = 0.65)
plt.xlabel("Vote Average",weight='bold')
plt.ylabel("Frequency",weight='bold')
plt.title("Histogram for Vote Average",weight='bold')
 plt.show()
```

Output :

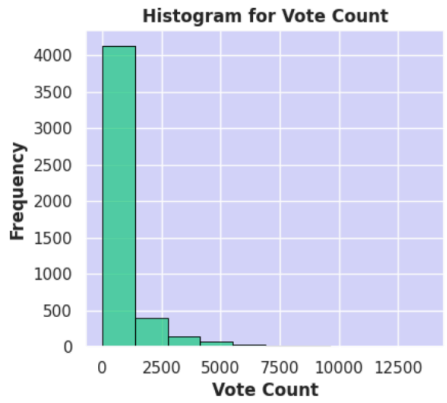


Histogram for Vote count :

code :

```
plt.figure(figsize = (4.5,4))
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white'})
sns.histplot(data = df , x = "vote_count" , bins = 10,
color ="#53c68c",edgecolor = "black" ,linewidth = 0.65 )
plt.xlabel("Vote Count",weight='bold')
plt.ylabel("Frequency",weight='bold')
plt.title("Histogram for Vote Count",weight='bold')
plt.show()
```

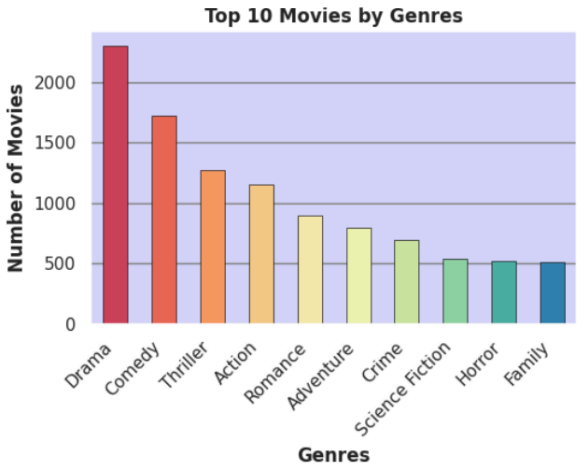
output :



Top 10 Movie Genres :
code :

```
genres_count = df["genres"].explode().value_counts()
genres_df = pd.DataFrame(genres_count).reset_index()
genres_count = df["genres"].explode().value_counts()
genres_df = pd.DataFrame(genres_count).reset_index()
plt.figure(figsize = (5.5,4.5))
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white' , "grid.color" : "grey"})
sns.barplot(x = genres_df["genres"].head(10),y = genres_df["count"].head(10),data = df ,
width = 0.5,edgecolor = "black" ,palette ="Spectral",linewidth = 0.45)
plt.xlabel("Genres",weight='bold')
plt.ylabel("Number of Movies",weight='bold')
plt.title("Top 10 Movies by Genres",weight='bold')
plt.xticks(rotation = 45 ,ha = "right")
plt.tight_layout()
plt.show()
```

output :

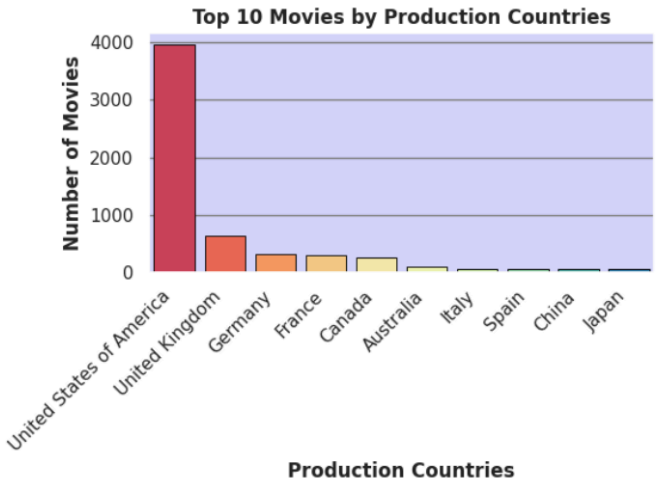


Top 10 Production Countries :
code :

```
countries_count = df["production_countries"].explode().value_counts()
countries_count_df = pd.DataFrame(countries_count).reset_index()
plt.figure(figsize = (6,4.5))
```

```
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white' , "grid.color" : "grey"})
sns.barplot(x = countries_count_df["production_countries"].head(10),
y = countries_count_df["count"].head(10), data = df , linewidth = 0.65, palette="Spectral",
edgecolor = "black" )
plt.xlabel("Production Countries",weight='bold')
plt.ylabel("Number of Movies",weight='bold')
plt.title("Top 10 Movies by Production Countries",weight='bold')
plt.xticks(rotation = 45 ,ha = "right")
plt.tight_layout()
plt.show()
```

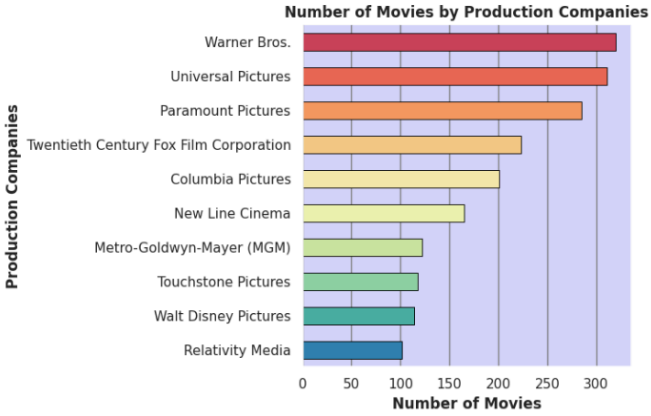
output :



Top 10 Production Companies :
code :

```
company_count = df["production_companies"].explode().value_counts()
company_count_df = pd.DataFrame(company_count).reset_index()
plt.figure(figsize = (7.5,5))
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white', "grid.color" : "grey"})
sns.barplot(x = company_count_df["count"].head(10),
data = df ,y=company_count_df["production_companies"].head(10),
palette="Spectral",edgecolor='black',width = 0.5,linewidth=0.65)
plt.ylabel("Production Companies",weight='bold')
plt.xlabel("Number of Movies",weight='bold')
plt.title("Number of Movies by Production Companies",weight='bold')
plt.tight_layout()
plt.show()
```

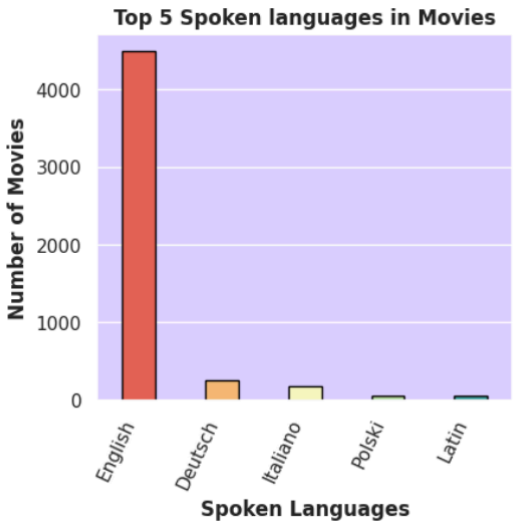
output :



Top 5 Spoken languages :
code :

```
language_count = df["spoken_languages"].explode().value_counts()
language_df = pd.DataFrame(language_count).reset_index()
language_df = language_df[language_df["spoken_languages"].str.match(r"[a-zA-Z]+$")]
plt.figure(figsize = (4.5,4))
sns.set(rc = {"axes.facecolor" : "#dacffc" , "figure.facecolor" : "white"})
sns.barplot(x = language_df["spoken_languages"].head(5) , y =language_df["count"].head(5) ,
palette= "Spectral",edgecolor ="black",width = 0.4)
plt.xlabel("Spoken Languages",weight='bold')
plt.ylabel("Number of Movies",weight='bold')
plt.title("Top 5 Spoken languages in Movies",weight='bold')
plt.xticks(rotation = 65 ,ha = "right")
plt.show()
```

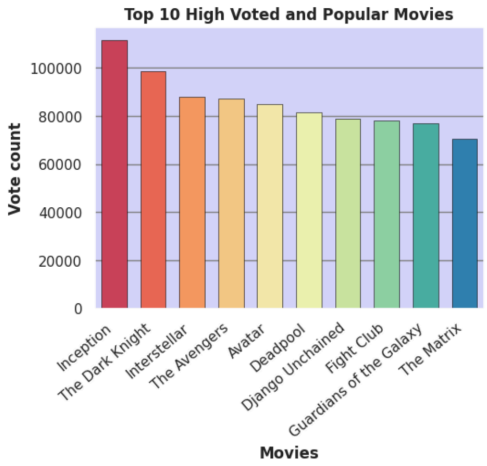
output :



Top 10 Popular Movies :
code :

```
vote_df = pd.DataFrame(df[["title","vote_average","vote_count"]])
vote_df["vote"] = vote_df["vote_average"]*vote_df["vote_count"]
vote_df = vote_df.sort_values("vote",ascending = False)
plt.figure(figsize = (5.5,4))
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white' , "grid.color" : "grey"})
sns.barplot(x = vote_df["title"].head(10) , y = vote_df["vote"].head(10) , data = vote_df ,
edgecolor = "black", palette = "Spectral",width = 0.65,linewidth = 0.4)
plt.xlabel("Movies",weight='bold')
plt.ylabel("Vote count",weight='bold')
plt.title("Top 10 High Voted and Popular Movies",weight='bold')
plt.xticks(rotation =40 , ha = "right")
plt.show()
```

output :



2.Bivariate Analysis :

Bivariate analysis is the simultaneous analysis of two variables. It explores the concept of the relationship between two variable whether there exists an association and the strength of this association or whether there are differences between two variables and the significance of these differences**.**

common types of graphs used in bivariate analysis :

Scatter Plot:

Displays individual data points on a two-dimensional graph, with one variable on the x-axis and the other on the y-axis. It is useful for identifying patterns, correlations between the two variables.
Example: A scatter plot showing the relationship between hours studied and exam scores.

Line Graph:

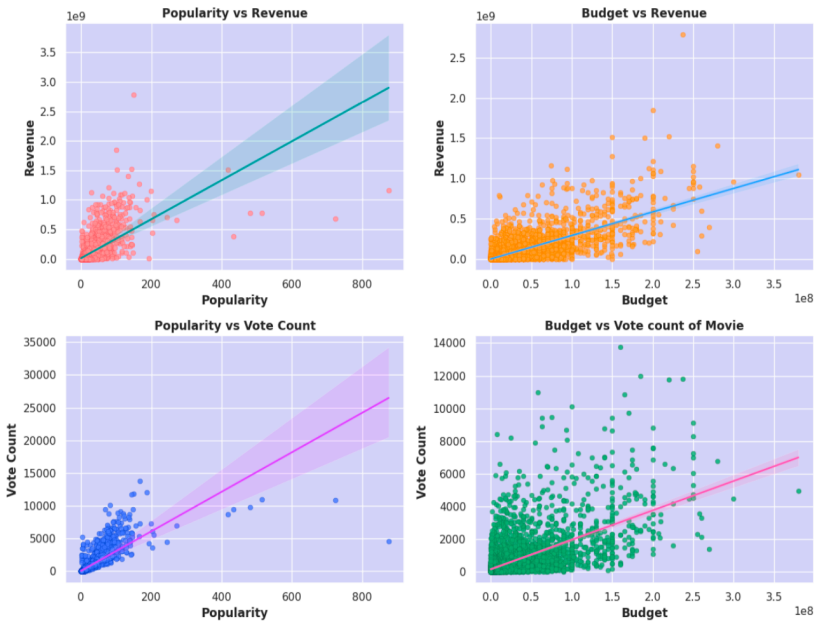
Often used to display the relationship between two variables over time. Multiple lines can be used to compare different groups .

Scatter plot for Bivariate analysis :

code :

```
# Subplot 1
# Scatter plot for Popularity and Revenue
sns.regplot(x="popularity", y="revenue", data=df ,color = "#ff9999" ,
        scatter_kws={'edgecolor': '#ff5050','linewidths': 0.3,'s' : 25},
        line_kws={'color': '#009999','linewidth':1.8}, ax=axes[0][0])
axes[0][0].set_xlabel("Popularity", fontweight='bold',fontsize = 12)
axes[0][0].set_ylabel("Revenue", fontweight='bold',fontsize = 12)
axes[0][0].set_title("Popularity vs Revenue", fontweight='bold',fontsize =12)

# Subplot 2
# Scatter plot for Budget vs Revenue
sns.regplot(x = "budget", y = "revenue" , data = df ,color = "#ffa64d" ,
        scatter_kws={'edgecolor': '#ff8000','linewidths': 0.3,'s' : 25},
        line_kws={'color': '#4da6ff','linewidth':1.8}, ax=axes[0][1])
axes[0][1].set_xlabel("Budget", fontweight='bold',fontsize = 12)
axes[0][1].set_ylabel("Revenue", fontweight='bold',fontsize = 12)
axes[0][1].set_title("Budget vs Revenue", fontweight='bold',fontsize =12)

# Subplot 3
# Scatter plot for Popularity vs Vote Count
sns.regplot(x = "popularity", y = "vote_count" , data = df ,color = "#4d79ff" ,
        scatter_kws={'edgecolor': '#002db3','linewidths': 0.3,'s' : 25},
        line_kws={'color': '#d24dff','linewidth':1.8}, ax=axes[1][0])
axes[1][0].set_xlabel("Popularity", fontweight='bold',fontsize = 12)
axes[1][0].set_ylabel("Vote Count", fontweight='bold',fontsize = 12)
axes[1][0].set_title("Popularity vs Vote Count", fontweight='bold',fontsize =12)

# Subplot 4
# Scatter plot for Budget vs Vote count of Movie
sns.regplot(x = "budget", y = "vote_count" ,data = df ,color = "#39ac73" ,
        scatter_kws={'edgecolor': '#26734d','linewidths': 0.3,'s' : 25},
        line_kws={'color': '#ff66b3','linewidth':1.8}, ax=axes[1][1])
axes[1][1].set_xlabel("Budget", fontweight='bold',fontsize = 12)
axes[1][1].set_ylabel("Vote Count ", fontweight='bold',fontsize = 12)
axes[1][1].set_title("Budget vs Vote count of Movie", fontweight='bold',fontsize = 12)

# Adjust layout for better display
plt.tight_layout()
plt.show()
```

output :



Multivariate data analysis :

Multivariate data refers to datasets where each observation or sample point consists of multiple variables or features. These variables can represent different aspects, characteristics, or measurements related to the observed phenomenon. When dealing with three or more variables, the data is specifically categorized as multivariate.

Scatter Plot Matrix (Pair Plot):

- A grid of scatter plots showing the relationship between each pair of variables.
- Diagonal elements often show a histogram or kernel density plot of individual variables.
- Useful for identifying patterns, correlations, and potential outliers.
- A pair plot, also known as a scatterplot matrix, is a matrix of graphs that enables the visualization of the relationship between each pair of variables in a dataset.
- It combines both histogram and scatter plots, providing a unique overview of the dataset's distributions and correlations.

Heatmaps :

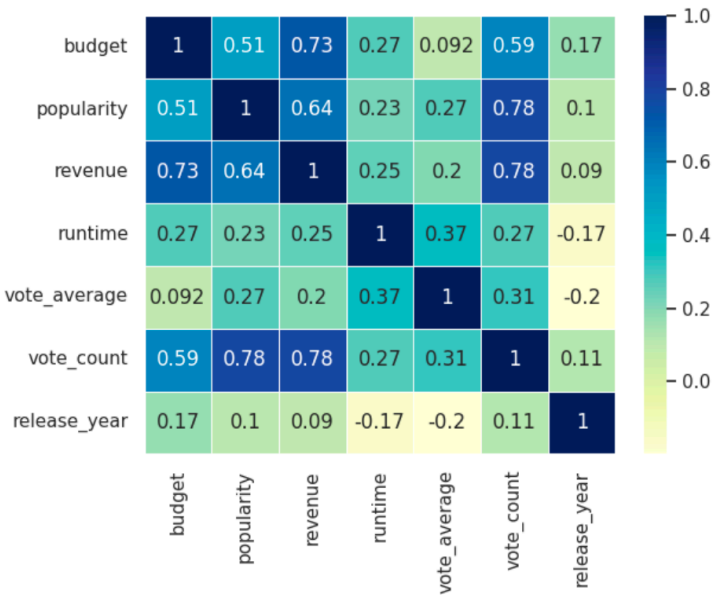
- A matrix where values are represented by colors.
- Useful for displaying the correlation matrix of variables or for visualizing data points across two categorical dimensions.
- A Heat map is a graphical representation of multivariate data that is structured as a matrix of columns and rows. Heat maps are very useful in describing correlation among several numerical variables, visualizing patterns and anomalies.

Heat Map for Multivariate Analysis :

code :

```
numerical_data = pd.DataFrame(df)
numerical_data.drop(columns =["genres","keywords","original_language",
                        "original_title","overview","production_companies","production_countries",
                        "release_date","spoken_languages","status","title",
                        "cast","crew","id"],inplace = True)
corr_matrix = numerical_data.corr()
sns.heatmap(corr_matrix ,annot = True,linewidths=.5 ,cmap="YlGnBu")
plt.show()
```

output :

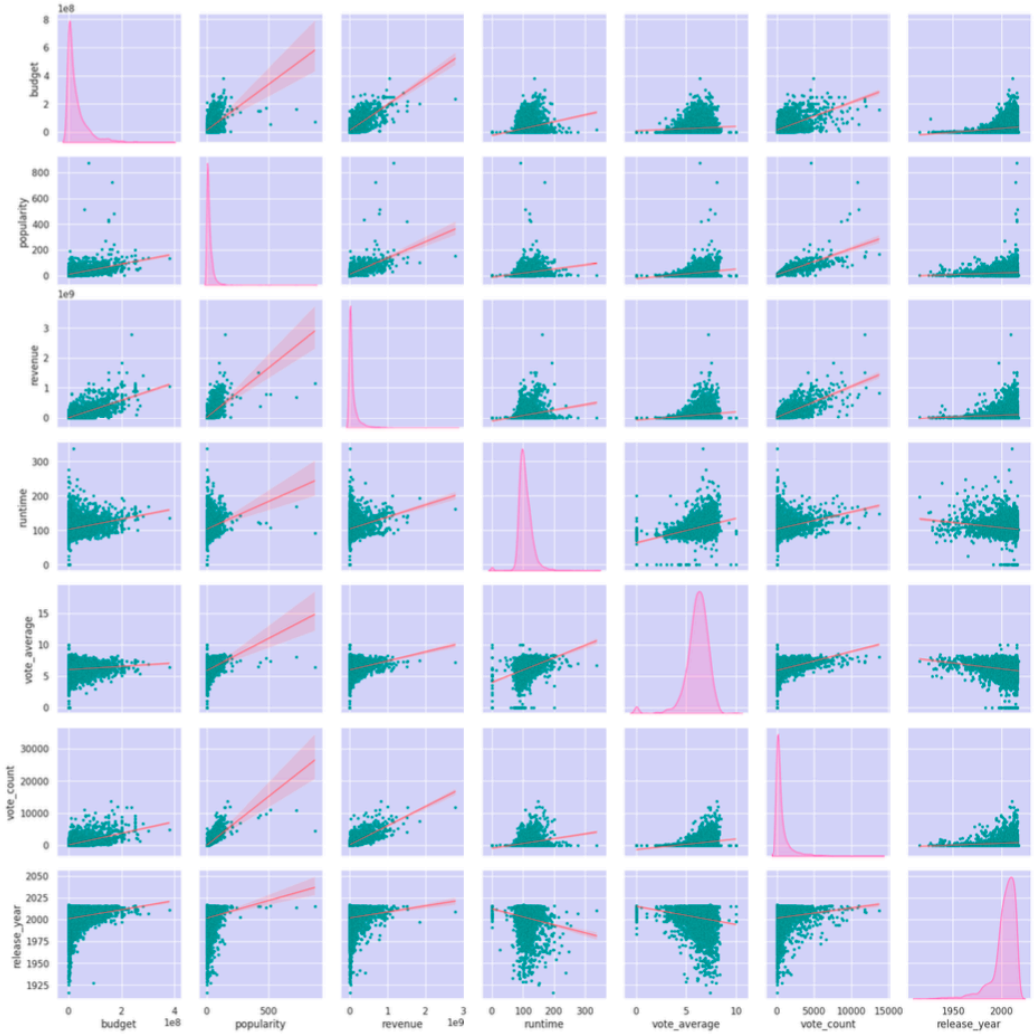


Pair plot :

code :

```
plt.figure(figsize = (10,10))
sns.set(rc={'axes.facecolor':'#d4d4f7', 'figure.facecolor':'white' , "grid.color" : "white"})
sns.pairplot(data = numerical_data ,diag_kind="kde",diag_kws = {"color": "hotpink"},kind="reg",
        plot_kws={'color':'#00b3b3','scatter_kws': {'s': 6.5 ,'edgecolor': 'darkcyan' },
        'line_kws':{'color':'#ff4d4d', 'linewidth':0.95}} )
plt.show()
```

output :



Conclusion :

To build a content based personalized movie recommendation system ,the Exploratory Data Analysis (EDA) process is very important and necessary step in analysing the dataset . The process involves analyzing and summarizing the main characteristics of the dataset, often using visual methods.By examining the relationships between different variables, EDA helps in identifying which features are most relevant for the recommendation model. Thus by performing EDA on the dataset the appropriate algorithm for building the recommendation system is selected for developing enhanced recommendation system.