Project - 2

**Uncovering suprising Facts from World Wide Movie Database using Data Cleaning & Data Visualization**

**OVERVIEW**

A project to overlook at the movie’s database and interpret various finding using Data cleaning, Data wrangling and Data Visualization

**Software Requirements**

1. Programming Language : Python

2. Environemnt: Jupyter Notebooks / Google Collab

3. Database: CSV(export type)

4. Operation System: Windows XP or above

5. Librarires Used: Pandas,Folium, Seaborn, Scikit, SKLEARN, Wordcount

6.Datasets used: TMDB Dataset

1. **Open a New Notebook and import the required libraires and read the csv file**

|  |  |
| --- | --- |
|  | import numpy as np  import pandas as pd  pd.set\_option('max\_columns', None)  import matplotlib.pyplot as plt  import seaborn as sns  %matplotlib inline  plt.style.use('ggplot')  import datetime  from scipy import stats  from wordcloud import WordCloud  from collections import Counter  from nltk.corpus import stopwords  from nltk.util import ngrams  import nltk  nltk.download('stopwords')  stop = set(stopwords.words('english'))  import os  import plotly.offline as py  py.init\_notebook\_mode(connected=True)  import plotly.graph\_objs as go  import plotly.tools as tls  from PIL import Image |

Description:

Importing all the necessary modules and an additional line is written i.e

Max columns , None this is done in order to avoid the limited column size

1. **Loading the training & testing Dataset**

train = pd.read\_csv('/data.csv')

Description:

Directly poviding the data from our computer memory and the parameter passed is the path of the file containg the data

1. **Visualizing the Distribution of Revenue with & without Log**

|  |  |
| --- | --- |
| fig, ax = plt.subplots(figsize=(16,6))  plt.subplot(1, 2, 1)  sns.distplot(train['revenue'], kde=False);  plt.title('Distrinution o frevenue');  plt.subplot(1, 2, 2)  sns.distplot(np.log1p(train['revenue']), kde=False);  plt.title('Distribution of log revenue')  Output:  Description:  Since A huge amount of data is to be represented the graph is compresed and does not provide an accurate result for analysis ,hence the logarithamic graph is taken so the values are reduced and provides more accurate result for analysis. Subplot is used when we need to plot multiple graphs in single output and the three parameters for it are the number of rows ,number of columns ,position of the graph.   1. **Finding the Relationship between Movie Revenue & Budget**   train['log\_revenue'] = np.log1p(train['revenue'])  train['log\_budget'] = np.log1p(train['budget'])  plt.figure(figsize=(16, 8))  plt.subplot(1, 2, 1)  sns.scatterplot(train['budget'], train['revenue'])  plt.title('Revenue vs budget');  plt.subplot(1, 2, 2)  sns.scatterplot(train['log\_budget'], train['log\_revenue'])  plt.title('log transfromation of revenue vs budget');  Description:  Scatter plot of both revenue vs Budget and log of revenue vs Budget is taken but the logarithamic scatterplot is more accurate for anlysis since there are some low Budget movies which have a high revenue  **Output:** |  |

1. **Impact of Film’s Revenue with or without Homepage**

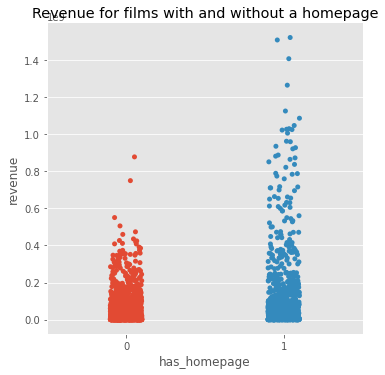
train['has\_homepage'] = 0

train.loc[train['homepage'].isnull() == False, 'has\_homepage'] = 1

sns.catplot(x='has\_homepage', y='revenue', data=train);

plt.title('Revenue for films with and without a homepage');

Output:

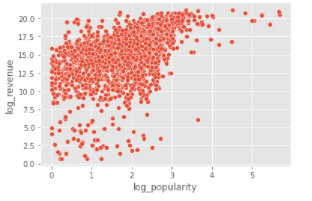
****

Description:

From the above graph we can analyse that the revenue of the movie is independent of the home page ,in the above graph until 0.4 revenue both are similar so we can say that the revenue is not entirely dependent on the home page except for the high Budget movies

1. **Relationship between Revenue and popularity**
2. data['log\_revenue'] = np.log1p(data['revenue'])
3. data['log\_popularity'] = np.log1p(data['popularity'])
4. sns.scatterplot(data['log\_popularity'],data['log\_revenue'])

Output:



Description:

The graph between revenue and popularity is plotted using the scatterplot and the revenue and the popularity are logarithamic with revenue on Y-axis and popularity on X-axis

1. **Frequent Words in Movie Titles**

plt.figure(figsize=(12, 12))

text =  ' '.join(train['original\_title'].values)

wordcloud = WordCloud(max\_font\_size=None,

                     background\_color ='white',

                     width =1200, height =1000).generate(text)

plt.imshow(wordcloud)

plt.title('Top word across movie titles')

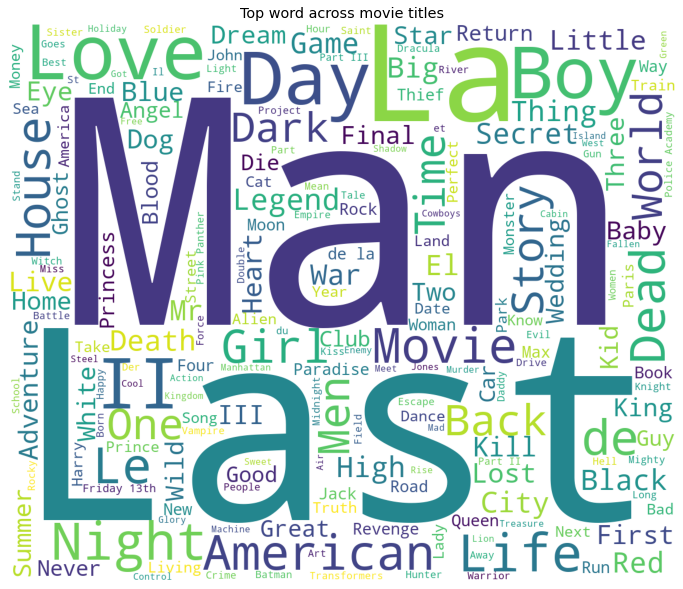
plt.axis('off')

plt.show()

Description:

Frequently used words in the movie titles is displayed using the word cloud ,here the higher occuring Word has greater size compared to other words

**Output:**

****

1. **Frequent Words in Movie Overviews**

plt.figure(figsize=(12, 12))

text =  ' '.join(train['overview'].fillna('').values)

wordcloud = WordCloud(max\_font\_size=None,

                     background\_color ='white',

                     width =1200, height =1000).generate(text)

plt.imshow(wordcloud)

plt.title('Top word across movie overviews')

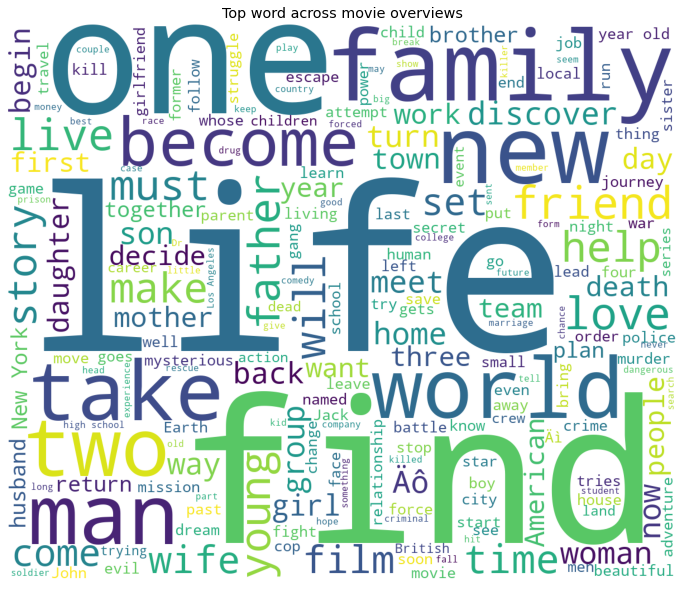
plt.axis('off')

plt.show()

Description:

The frequently used words can be obtained using the Word cloud which tells the most occuring words in the specified feild and greater size of the Word indicates the more occurence of the word

**Output:**

****

1. **Most frequent genres:**

plt.figure(figsize=(12, 12))

text =  ' '.join(data['all\_genres'].fillna('').values)

wordcloud = WordCloud(max\_font\_size=None,

                     background\_color ='white',

                     width =1200, height =1000).generate(text)

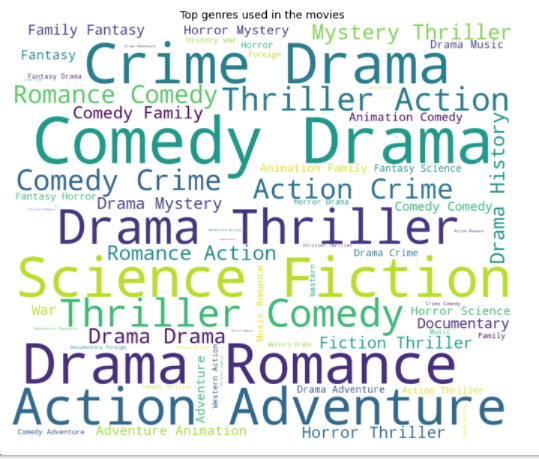
plt.imshow(wordcloud)

plt.title(' Top genres used in the movies’)

plt.axis('off')

plt.show()

**Output:**

****

Description:

Frequent genres is displayed using the Word cloud

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |