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:

The above part of the program is having a specific operation of importing various libraries/modules for the python program to work as expected.

“NumPy” is used to compute mathematical operations in python.

“pandas” is used to deal with datasets.

“matplotlib.pyplot” is used to display various types of graphical data.

“seaborn” acts as a high-level interface for visualization of the data in python.

“datetime” module is used to deal with date & time in python.

“wordcloud” is used to represent any data in the form of word cloud.

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

train = pd.read\_csv('/data.csv')

Description:

The above part of the program is having an operation of storing the csv data of the file data.csv in the variable named train.

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:  The above part of the code is having a specific operation i.e.,  “fig, ax = plt.subplots(figsize=(16,6))” is used to store width and height in variables fig and ax  “plt.subplot(1, 2, 1)” is used to identify total visualizations i.e., for subplot(1,2,1) represents there will be 1 row and 2 columns of data to be visualized and other 1 acts as an index of the data to be represented.  “sns.distplot(train['revenue'], kde=False);” is used to plot the data w.r.t the attribute named “revenue”.  .title is used to keep a proper title for the data to be represented.  “plt.subplot(1, 2, 2)” is used to represent there is an other data to be visualized of 1 row and 2 columns and other 2 acts as an index of the data to be represented.  “sns.distplot(np.log1p(train['revenue']), kde=False);” is used to plot the data w.r.t to the attribute “revenue” and the data is plotted using log function which makes more sense in visualization of the data.   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:  The above piece if the code is used to represent the data between movie revenue and budget.  “train['log\_revenue'] = np.log1p(train['revenue'])  train['log\_budget'] = np.log1p(train['budget'])”  is used to store the log values of attribute “revenue” in variable train[‘log\_revenue’] and log value of attribute “budget” is stored in variable train[‘log\_budget’].  Now the data is represented using scattered plotting for revenue and budget and also for log values of revenue and budget to reduce the disturbance in the data.  **Output:** |  |

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

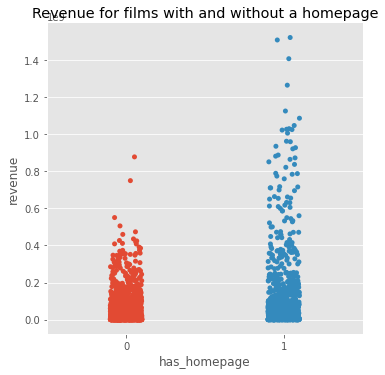
train['has\_homepage'] = 0

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:

The above part of the code is representing the impact of the film’s revenue with or without homepage.

“train['has\_homepage'] = 0” is written to initially keep the value of train[‘has\_homepage’] as 0.

“train.loc[train['homepage'].isnull() == False, 'has\_homepage'] = 1” is written if any of the film have homepage then it will update to 1 else 0.

“sns.catplot(x='has\_homepage', y='revenue', data=train);” is used to plot the data in cat plot method by keeping the x-axis as ‘has\_homepage’ and y-axis as ‘revenue’ and representing the data.

“plt.title('Revenue for films with and without a homepage');” is used to keep the title of the data represented as ‘Revenue for films with and without a homepage’.

1. **Films Revenue in various Languages**

language\_data = train.loc[train['original\_language'].isin(train['original\_language'].value\_counts().head(10).index)]

plt.figure(figsize=(16,8))

plt.subplot(1, 2, 1)

sns.boxplot(x='original\_language', y = 'revenue', data=language\_data )

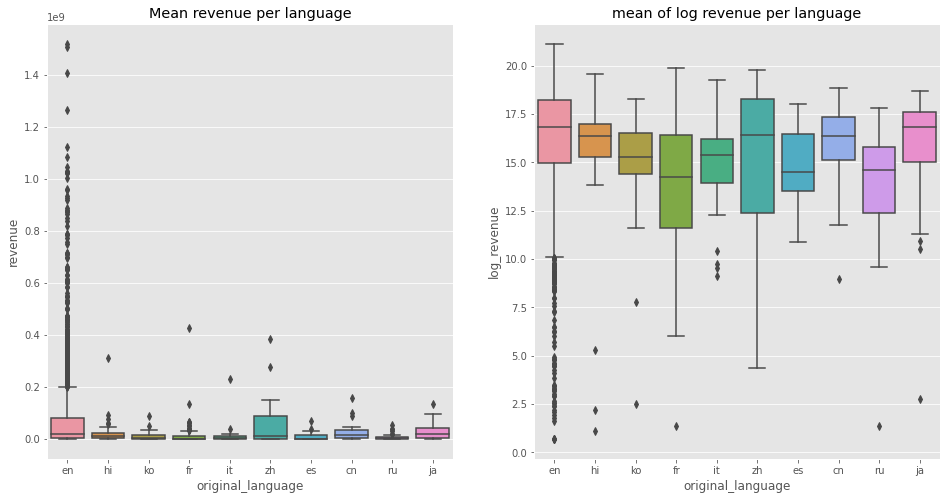
plt.title('Mean revenue per language')

plt.subplot(1, 2, 2)

sns.boxplot(x='original\_language', y = 'log\_revenue', data=language\_data)

plt.title('mean of log revenue per language')

Output:



Description:

The above part of the code represents the revenues of the film in various languages.

“language\_data = train.loc[train['original\_language'].isin(train['original\_language'].value\_counts().head(10).index)]” is used to store all the movies according to “original\_language” in “language\_data”.

“plt.figure(figsize=(16,8))

plt.subplot(1, 2, 1)

sns.boxplot(x='original\_language', y = 'revenue', data=language\_data )

plt.title('Mean revenue per language')”

the above lines are used to mention the width and height of the graphs; Rows, Columns and Index of the Visualization by keeping x-axis as ‘original\_language’ and y-axis as ‘revenue’ plotting ‘language\_data’ in box plot method by keeping the title of the data as ‘Mean revenue per language’.

“plt.subplot(1, 2, 2)

sns.boxplot(x='original\_language', y = 'log\_revenue', data=language\_data)

plt.title('mean of log revenue per language')”

the above lines of code are to mention the index number as 2 and by plotting the same graph except for changing y-axis from ‘revenue’ to ‘log\_revenue’ which represents the data properly and changing the title accordingly.

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:

The above part of the code is having significance importance in finding out the frequent words in movie titles.

“plt.figure(figsize=(12, 12))” is used to represent the width and height of the data to be represented.

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

                     background\_color ='white',

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

plt.imshow(wordcloud)

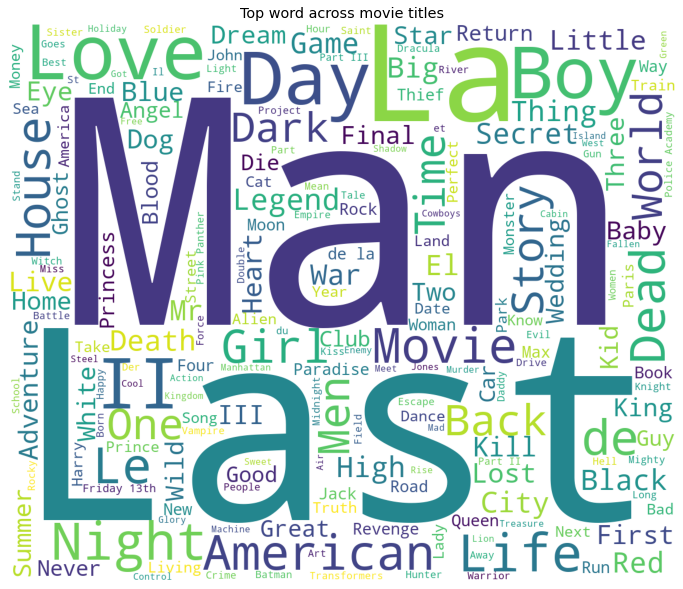
plt.title('Top word across movie titles')

plt.axis('off')

plt.show()”

The above lines are used to in converting all the words to wordcloud from which we can identify the frequency of the words by checking how large it is and displaying the wordcloud by keeping the title as “Top word across movie titles”.

**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 above part of the code is having significance importance in finding out the frequent words in movie titles.

“plt.figure(figsize=(12, 12))” is used to represent the width and height of the data to be represented.

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

                     background\_color ='white',

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

plt.imshow(wordcloud)

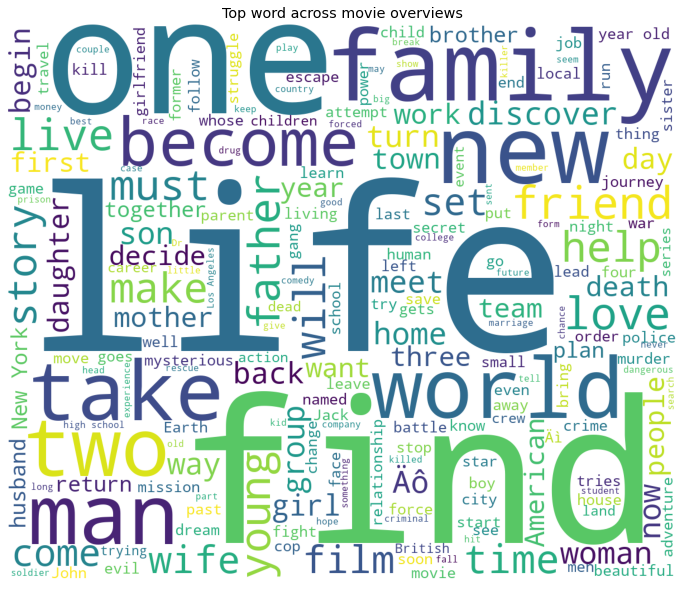
plt.title('Top word across movie overviews')

plt.axis('off')

plt.show()

The above lines are used to in converting all the words to wordcloud from which we can identify the frequency of the words by checking how large it is and displaying the wordcloud by keeping the title as “Top word across movie overviews”.

**Output:**

****

|  |
| --- |
|  |

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

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