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 the libraries for data visualisation, exploratory data analysis, statistical calculations, Word cloud.

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

train = pd.read\_csv('/data.csv')

**Description**: converts csv file file to dataframe format.

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:s  **Description**: Since, data is big. In plot1, data gets mixed up and data is not visualised properly. In plot2, using logarithmic data, data is properly visualised. In above 2 plots, plot2 is the best visualised.   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**: In graph1, low Budget movies got low revenue, high Budget movies got high revenue. In graph2, there are movies which get high revenue for low Budget, and high Budget movie got high revenue. Hence, graph2 is the best.  **Output:** |  |

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

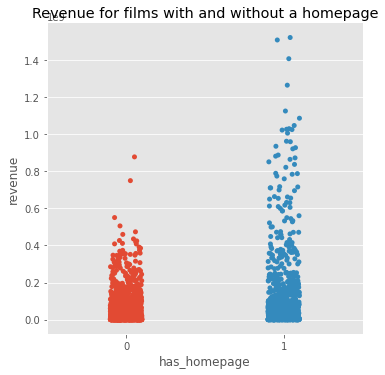
train['has\_homepage'] = 0

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 revenue is not dependent is on homepage except for the high Budget films.

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')

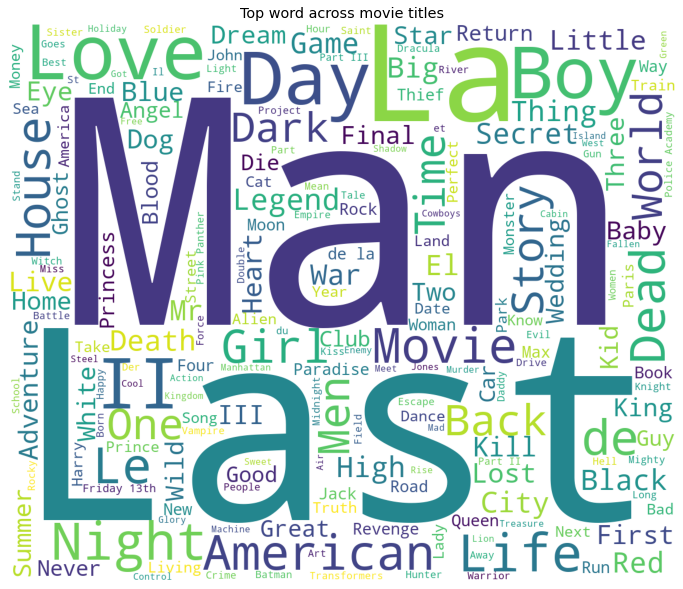
plt.axis('off')

plt.show()

**Description**: A word cloud is a collection or cluster of words decipated in different sizes.

The **titles** which are more frequent will be bold and bigger. The **titles** which are less frequently used will be smaller in size and thinner.

**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)

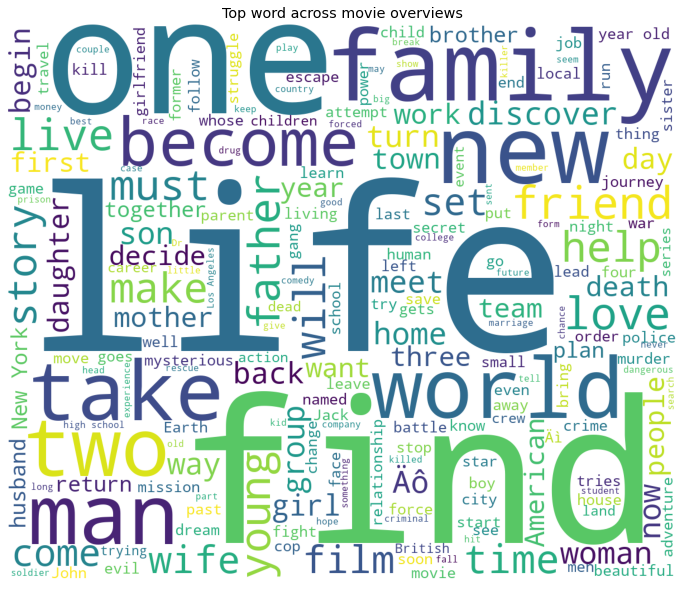
plt.title('Top word across movie overviews')

plt.axis('off')

plt.show()

**Description:** The words which are more frequently used in movie overviews will be bigger in size and thicker. The words which are less frequently used in movie overviews will be smaller in size and thinner.

**Output:**

****

**Conclusion:**

Imported few important libraries to manipulate a clean dataset, data visualisation, exploratory data analysis, statistical calculations. Later we collected data in csv format, converted csv file to data frames, visualised the data by using seaborne and analysed which graph is best, compared normal and logarthmic applied records.

|  |
| --- |
|  |

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

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