Gishnavi Kolli

Box office Data Analysis & Interpretation

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
import seaborn as s
import matplotlib.pyplot as plt
pd.set_option('max_columns', None)
%matplotlib inline
plt.style.use('ggplot')
plt.style.use('tableau-colorblind10')
import datetime
import pandas.util.testing as tm
from scipy import stats
from scipy.sparse import hstack, csr_matrix
from sklearn.model_selection import train_test_split, KFold
from wordcloud import WordCloud
from collections import Counter
import nltk
from nltk.corpus import stopwords
from nltk.util import ngrams
nltk.download('stopwords')
stop = set(stopwords.words('english'))
import os
import plotly.tools as tls
from sklearn.preprocessing import StandardScaler
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
import json, ast
from urllib.request import urlopen
from PIL import Image
```

- NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms,and much more.

- pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with relational or labeled data both easy and intuitive. pandas is well suited for many different kinds of data: Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet.

- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a

brief introduction to the ideas behind the library, you can read the introductory notes.

- matplotlib. pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc

- ggplot is a Python implementation of the grammar of graphics.

- %matplotlib inline sets the backend of matplotlib to the 'inline' backend: With this backend, the output of plotting commands is displayed inline within frontends like the Jupyter notebook, directly below the code cell that produced it. The resulting plots will then also be stored in the notebook document.

- Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals

- SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely

- Sklearn and Model_selection is a Python library that offers various features for data processing that can be used for classification, clustering, and model selection. Model_selection is a method for setting a blueprint to analyze data and then using it to measure new data

- train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. ... By default, Sklearn train_test_split will make random partitions for the two subsets. However, you can also specify a random state for the operation.

- Wordcloud is a visual representation of text data. It displays a list of words, the importance of each beeing shown with font size or color. This format is useful for

quickly perceiving the most prominent terms

- The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language

- plotly is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific

- JavaScript Object Notation (JSON) is a standardized format commonly used to transfer data as text that can be sent over a network. It's used by lots of APIs and Databases, and it's easy for both humans and machines to read. JSON represents objects as name/value pairs, just like a Python dictionary.

2. Loading the training & testing Dataset

```
# Importing the dataset
import io
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

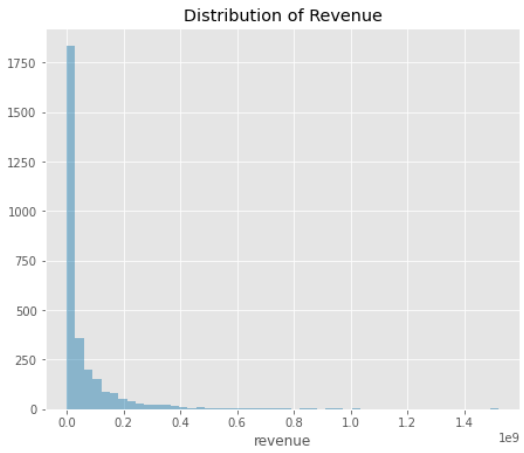
- we read the data from train.csv and store the data in train variable

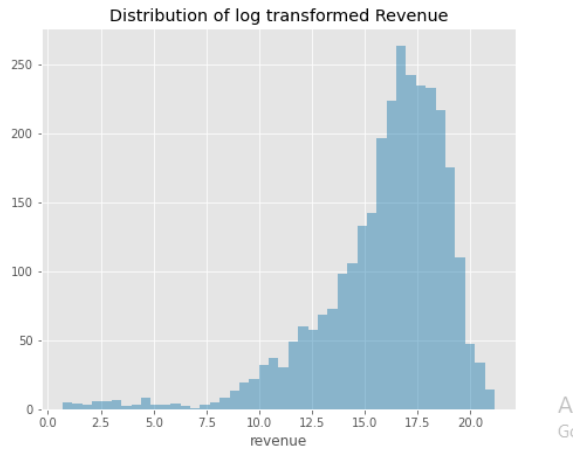
- we read the data from test.csv and store the data in testvariable

3. Visualizing the Distribution of Revenue with & without Log

```
fig, ax = plt.subplots(figsize=(16,6))
plt.subplot(1,2,1)
s.distplot(train['revenue'], kde= False);
plt.title('Distribution of Revenue');
plt.subplot(1,2,2)
s.distplot(np.log1p(train['revenue']), kde= False);
plt.title('Distribution of log transformed Revenue');
```

Output:





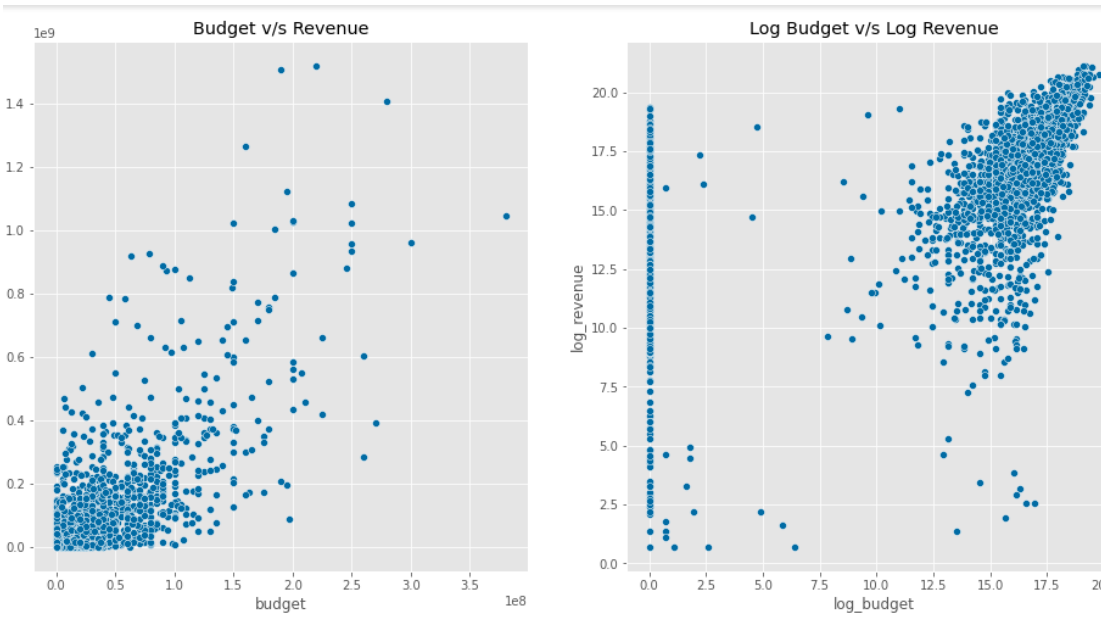
- subplots method provides a way to plot multiple plots on a single figure. Given the number of rows and columns , it returns a tuple (fig , ax), giving a single figure fig with an array of axes ax .
- Seaborn distplot shows a histogram with a line on it. ... We use seaborn in combination with matplotlib, the Python plotting module. A distplot plots a univariate distribution of observations
- plt.title title() method in matplotlib module is used to specify title of the visualization depicted and displays the title using various attributes.

.

4. Finding the Relationship between Movie Revenue & Budget

```
train['log_budget'] = np.log1p(train['budget'])
```

```
# Relationship between Film Revenue and Film Budget
plt.figure(figsize=(16,8))
plt.subplot(1,2,1)
s.scatterplot(train['budget'], train['revenue'])
plt.title('Budget v/s Revenue');
plt.subplot(1,2,2)
s.scatterplot(train['log_budget'], train['log_revenue'])
plt.title('Log Budget v/s Log Revenue');
```

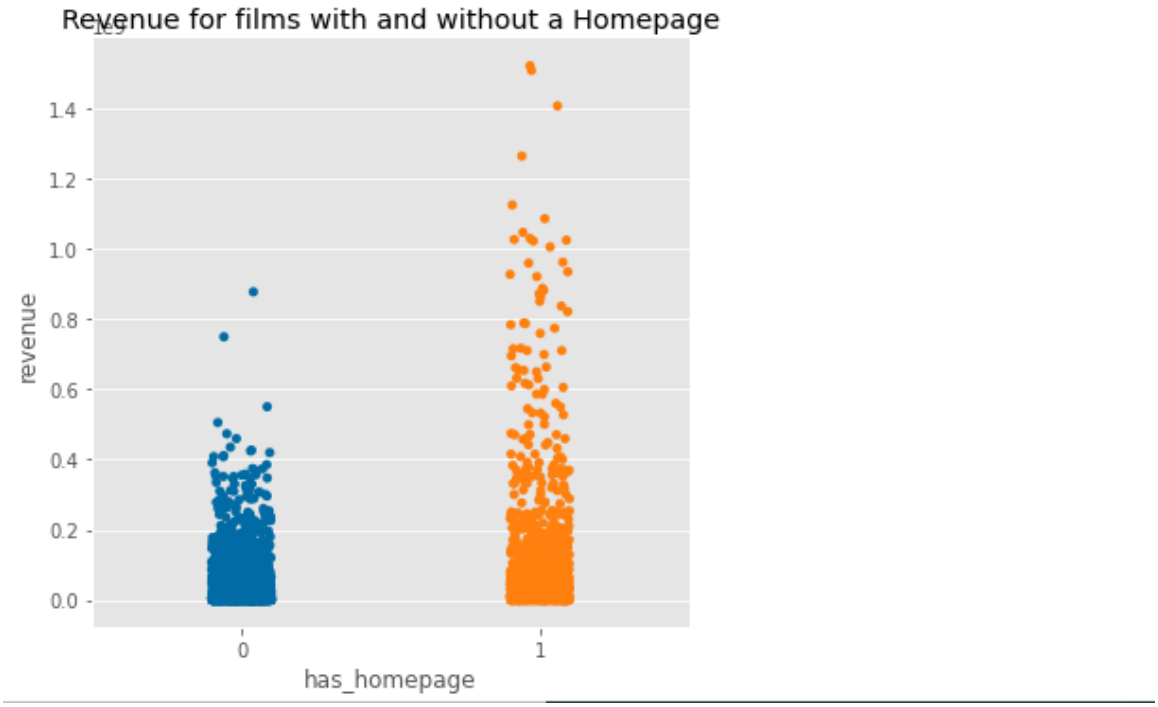


- Matplotlib allows the aspect ratio, DPI and figure size to be specified when the Figure object is created, using the figsize and dpi keyword arguments. figsize is a tuple of the width and height of the figure in inches, and dpi is the dots-per-inch (pixel per inch). To create an 800x400 pixel, 100 dots-per-inch figure
- subplots() without arguments returns a Figure and a single Axes
- plt.scatterplot will create a scatter plot for the budget and revenue from train dataset
- plt.title is used to set the title of the plot to budget v/s revenue
- s.scatterplot(train['log_budget'], train['log_revenue'])
- this will create a scatter plot for log budget and log revenue
- plt.title('Log Budget v/s Log Revenue');
- this will set the title as Log Budget v/s Log Revenue

5. Impact of Film's Revenue with or without Homepage

```
test['has_homepage'] = 0
test.loc[test['homepage'].isnull()== False, 'has_homepage'] = 1
```

```
s.catplot(x= 'has_homepage' , y= 'revenue' , data= train)
plt.title('Revenue for films with and without a Homepage');
```

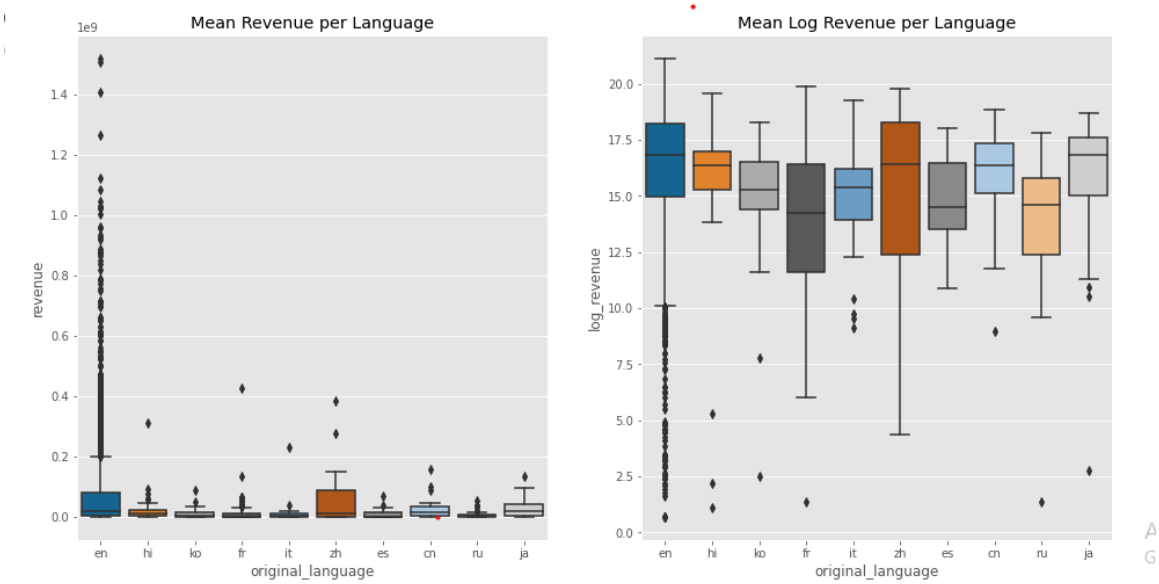


- for train data set set has_hompage equal to 0
- and loc method is used to return boolean value if train['homepage'] is null
- catplot shows frequencies (or optionally fractions or percents) of the categories of one, two or three categorical variables. it takes x and y axis and dataset as input
- plt.tyitle is used to set title as Revenue for films with and without a homepage

6. 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)
s.boxplot(x='original_language', y='revenue', data= language_data)
plt.title('Mean Revenue per Language');
plt.subplot(1,2,2)
s.boxplot(x='original_language', y='log_revenue', data= language_data)
plt.title('Mean Log Revenue per Language');
```

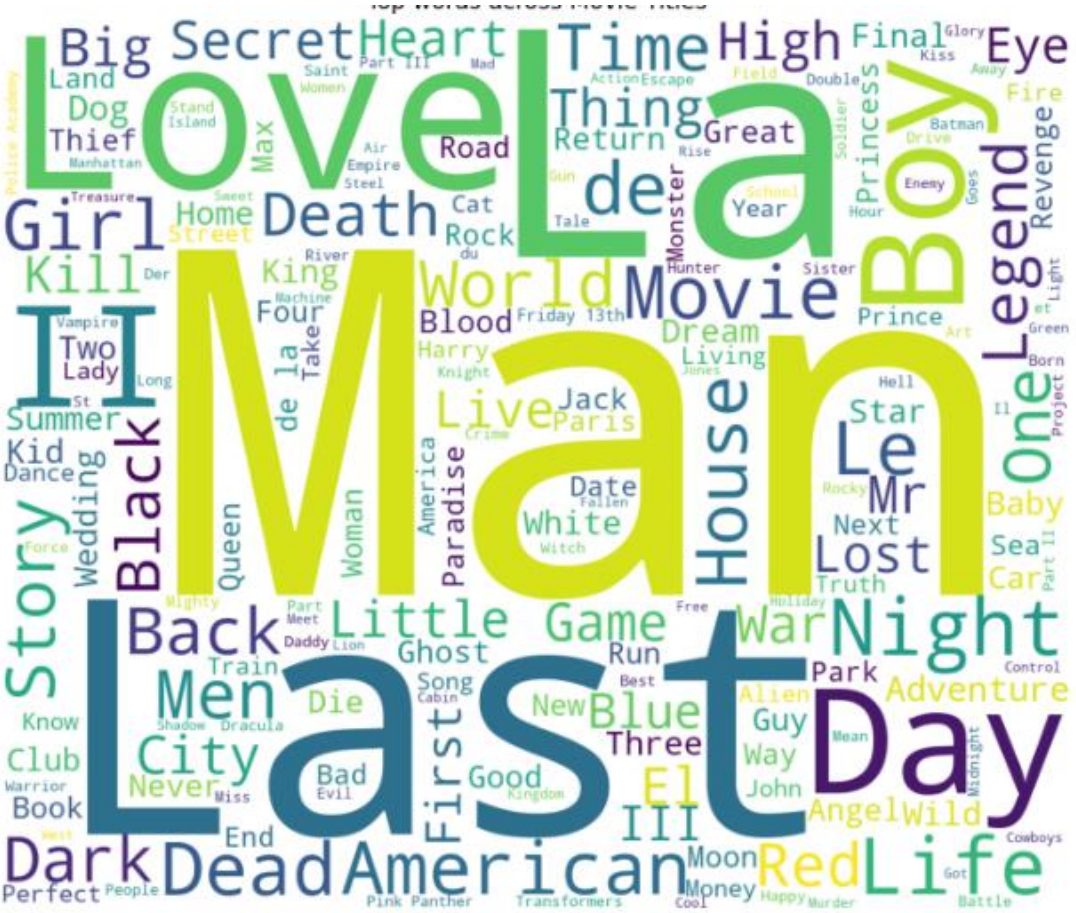


- plt.figure(figsize=(16,8))
- plt,figure is used to get the figure of desired size.
- The whole figure is regarded as the figure object. It is necessary to explicitly use plt.figure() when we want to tweak the size of the figure and when we want to add multiple Axes objects in a single figure.
- plt.subplot(1, 2, 1)
- The first two optional arguments of plt.subplots define the number of rows and columns of the subplot grid.
- s.boxplot(x='original_language', y = 'revenue', data=language_data)
- we find the boxplot of the data using s.boxplot and giving it x, y axis names and feed it data set
- plt.title('Mean revenue per language')
- plt.title('Mean revenue per language')
- plt. title is used to set title of the plot here Mean revenue per language
- plt.subplot(1, 2, 2)
- plt.subplot(1, 2, 2)

- The first two optional arguments of plt.subplots define the number of rows and columns of the subplot grid.
- s.boxplot(x='original_language', y='log_revenue', data= language_data)
- .boxplot(x='original_language', y = 'revenue', data=language_data)
- we find the boxplot of the data using s.boxplot and giving it x, y axis names and feed it data set
- plt.title('Mean Log Revenue per Language');
- plt. title is used to set the title of the plot here Mean Log Revenue per Language

7. 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 words across Movie Titles')
plt.axis('off')
plt.show()
```



- plt.figure(figsize= (12,12))
- plt,figure is used to get the figure of desired size.
- The whole figure is regarded as the figure object. It is necessary to explicitly use plt.figure() when we want to tweak the size of the figure and when we want to add multiple Axes objects in a single figure.
- text= ' '.join(train['original_title'].values)
- we use "".join takes all the iterable and joins them into one here original_title
- wordcloud= WordCloud(max_font_size= None,
- background_color= 'white',

- width= 1200, height=1000).generate(text)
- Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud.
- it needs max_font_size and background_color
- plt.imshow(wordcloud)
- imshow(I) displays the grayscale image I in a figure. imshow uses the default display range for the image data type and optimizes figure, axes, and image object properties for image display.
- plt.title('Top words across Movie Titles')
- plt. title is used to set title of the plot here Mean revenue per language
- plt.axis('off')
- plt.show()
- Turns off axes and shows the output

8. 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 Descriptions across Overviews')
plt.axis('off')
plt.show()
```

- plt.figure(figsize= (12,12))
- plt,figure is used to get the figure of desired size.
- The whole figure is regarded as the figure object. It is necessary to explicitly use plt.figure() when we want to tweak the size of the figure and when we want to add multiple Axes objects in a single figure.
- text= ' '.join(train['original_title'].values)
- we use "".join takes all the iterable and joins them into one here original_title
- wordcloud= WordCloud(max_font_size= None,
- background_color= 'white',

- width= 1200, height=1000).generate(text)
- Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud.
- it needs max_font_size and background_color
- plt.imshow(wordcloud)
- imshow(I) displays the grayscale image I in a figure. imshow uses the default display range for the image data type and optimizes figure, axes, and image object properties for image display.
- plt.title('Top Descriptions across Overviews')
- plt. title is used to set title of the plot here Mean revenue per language
- plt.axis('off')
- plt.show()
- Turns off axes and shows the output

conclusion:

Firstly, we imported the required libraires and read the csv file and Loaded the training & testing Dataset

then Visualized the Distribution of Revenue and plotted the Relationship between Movie Revenue & Budget

Impact of Films Revenue with or without Homepage has been plotted above and Films Revenue in various Languages

After that we found out the most frequently used movie title that is last, man, love etc. and we also found the most frequently used words in movie overviews those are life, find, world etc.

Therefore, we have successfully analyzed the movies database and interpret various finding using data cleaning, data wrangling and data visualization