DEPARTMENT OF COMPUTER ENGINEERING

**PRESIDENCY UNIVERSITY**

**Itgalpur Rajanakunte, Yelahanka, Bengaluru, Karnataka-5**

# Web scraping, Linear regression, and Sentiment analysis

***A mini project report submitted by***

**Monica K(20181COM0077)**
**Rishika PJ(20181COM0124)**

***As part of the lab based course CSE260***
***Introduction to Data science Lab***

**BACHELOR OF TECHNOLOGY**

***in***

**COMPUTER ENGINEERING & DATA SCIENCE**

***under the supervision of***

**Dr. S Suresh Kumar, Adjunct Faculty**

**April  2020**

# BONAFIDE CERTIFICATE

**This is to certify that the project report entitled, "Web scrapping, Linear regression and Sentiment analysis" is a Bonafede record of Mini Project work done as part of CSE260 Introduction to Data Science lab during the academic year 2020-2021 by**

**Monica K(20181COM0077)**
**Rishika PJ(20181COM0124)**

Submitted on – 15th May 2020

# CONTENTS

# 1.INTRODUCTION

Over the last decade there's been a massive explosion in both the data generated and retained by companies, as well as you and me. Sometimes we call this "big data," and like a pile of lumber we'd like to build something with it. Data scientists are the people who make sense out of all this data and figure out just what can be done with it. The helm of generating robust, actionable analytics from immense data sets. It's these efforts that bring clarity to how people interact with the web and are the basis for usable features that inform critical business strategies.

In this project report we have put together basic concepts of web scrapping, performing linear regression on scrapped data, implementing few data science techniques on scrapped data, and sentiment analysis. Each of these concepts are very important for a Data scientist in numerous ways. Knowledge of these few concepts along with many more makes it an easier and smarter way to understand the immense data being produced every day, every minute, every second. It helps companies take productive decisions which help the company in the long run.

# 2. WEB SCRAPPING

Web scrapping is a technique employed to to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format. Data displayed by most websites can only be viewed using a web browser. They do not offer the functionality to save a copy of this data for personal use. The only option then is to manually copy and paste the data - a very tedious job which can take many hours or sometimes days to complete. Web Scraping is the technique of automating this process, so that instead of manually copying the data from websites, the Web Scraping software will perform the same task within a fraction of the time. Our work for this topic is carried forward below.

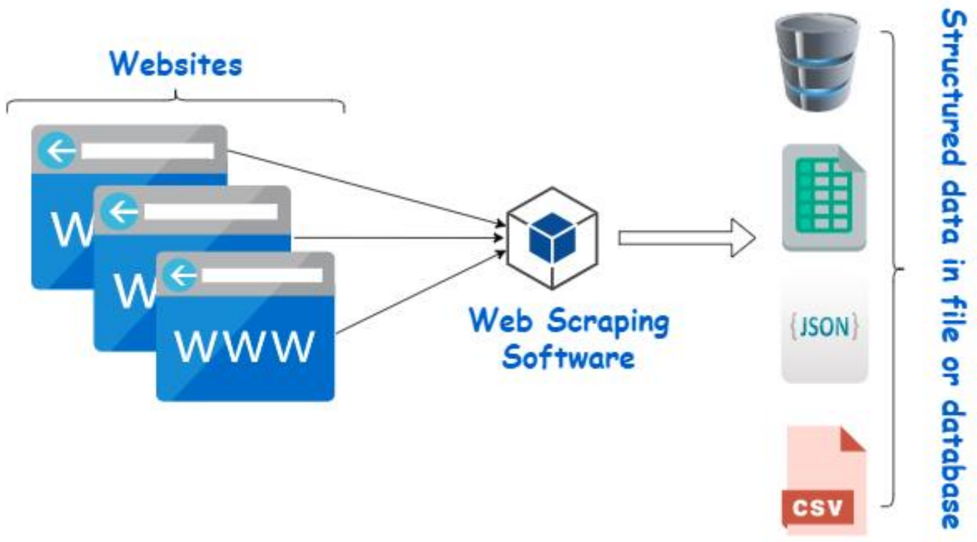
To help you get the whole picture, we will list each advantage and disadvantage of web scraping that we consider to be important.

**PROS**
Here are the advantages of data scraping.

- **Automation**
  Imagine how much time you would spend if you had to copy and paste each piece of information you need from a website. Not only would this take hours but it would drain all your energy. Luckily, scraping software automates most of the associated processes.

- **Accuracy**
  Not only is scraping fast but it is also extremely accurate. This prevents any major mistakes which can occur as a result of smaller data extraction mistakes made during the process.

- **Data management**
  You use spreadsheets and databases to manage figures and numerals on your computer, but you can’t really do this on a website configured in HTML. With web scraping tools, this is made possible.



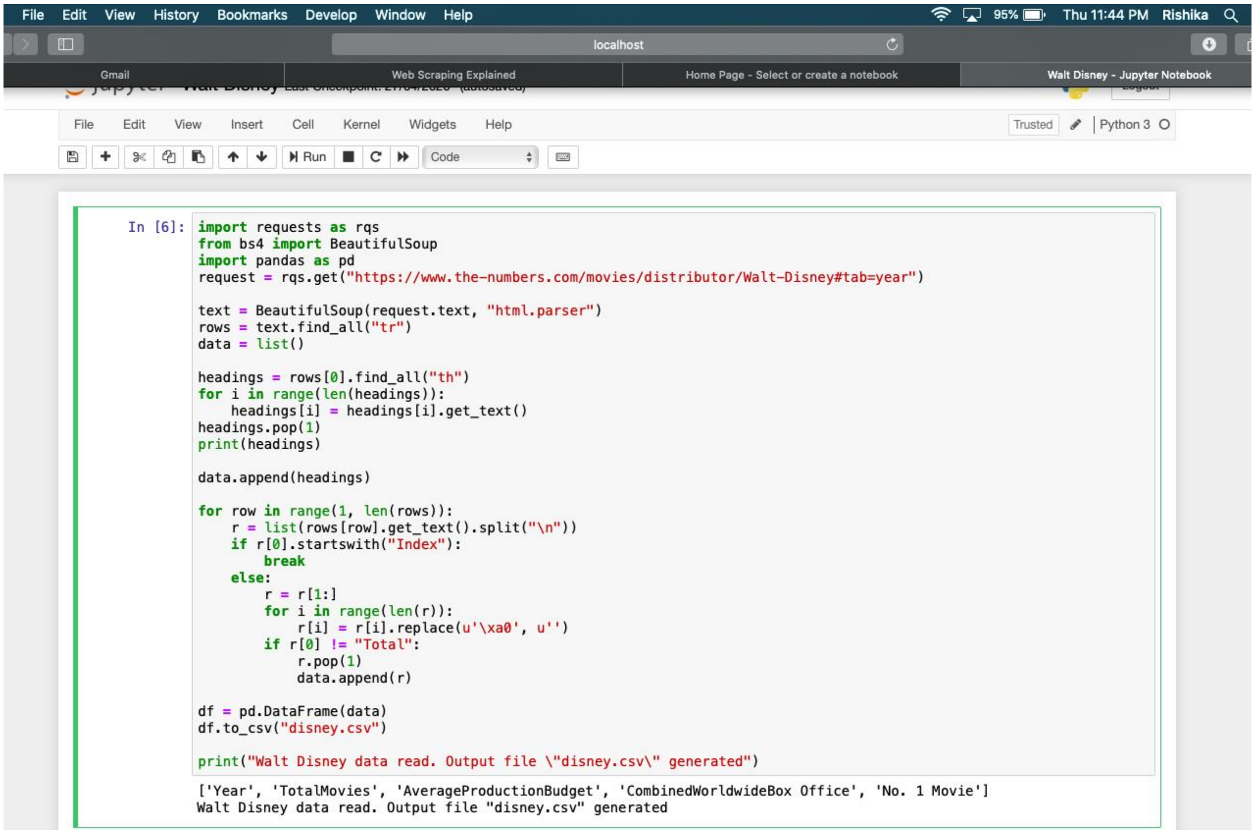
1. Find the URL that you want to scrape
2. Inspecting the Page
3. Find the data you want to extract
4. Write the code
5. Run the code and extract the data
6. Store the data in the required format

### 1.1 The concern

- Extract Walt Disney company's income data based on the popularity of the genre of the movie being preferred.
- Extract details such as Movie name and genre, along with income made by each genre.

## 1.2 Our solution

## CODE:



```
In [6]: import requests as rqs
from bs4 import BeautifulSoup
import pandas as pd
request = rqs.get("https://www.the-numbers.com/movies/distributor/Walt-Disney#tab=year")

text = BeautifulSoup(request.text, "html.parser")
rows = text.find_all("tr")
data = list()

headings = rows[0].find_all("th")
for i in range(len(headings)):
    headings[i] = headings[i].get_text()
headings.pop(1)
print(headings)

data.append(headings)

for row in range(1, len(rows)):
    r = list(rows[row].get_text().split("\n"))
    if r[0].startswith("Index"):
        break
    else:
        r = r[1:]
        for i in range(len(r)):
            r[i] = r[i].replace(u'\xa0', u'')
        if r[0] != "Total":
            r.pop(1)
            data.append(r)

df = pd.DataFrame(data)
df.to_csv("disney.csv")

print("Walt Disney data read. Output file \"disney.csv\" generated")
```

```
['Year', 'TotalMovies', 'AverageProductionBudget', 'CombinedWorldwideBox Office', 'No. 1 Movie']
Walt Disney data read. Output file "disney.csv" generated
```

- The output of the above obtained scraped data, is a CSV file. When downloaded, it is in the form of an excel sheet.

- Using this data, Data science techniques can be used to draw beneficial inferences for the Walt Disney company.

# 3.DATA SCIENCE TECHNIQUES IMPLEMENTED

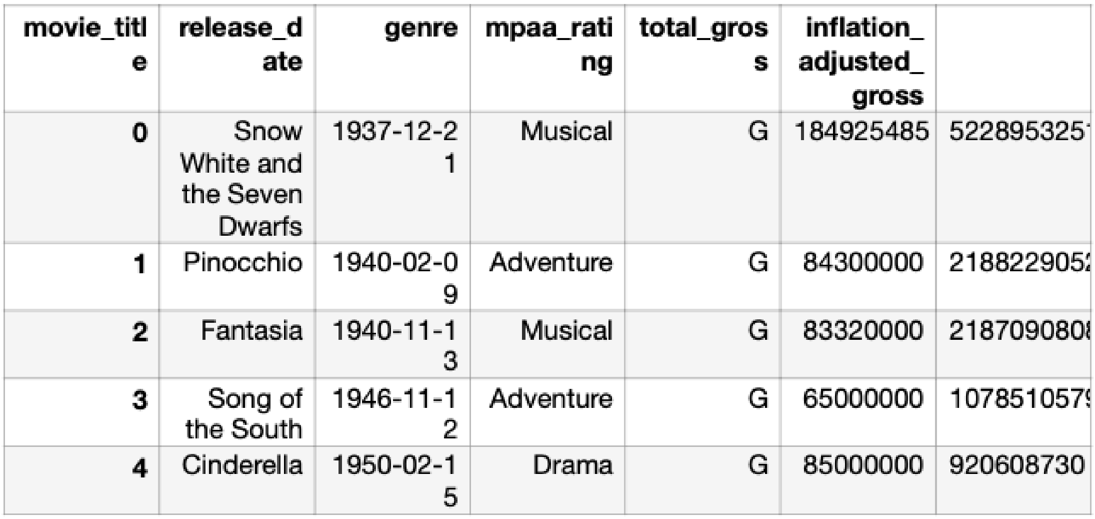
## 3.1 Data collection

- the process of **gathering** and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes.

- From the obtained data after scrapping, we use Pandas library in Python to load the data set to carry out further operations.

- Walt Disney Studios is the foundation on which The Walt Disney Company was built. The Studios has produced more than 600 films since their debut film, Snow White and the Seven Dwarfs in 1937. While many of its films were big hits, some of them were not. In this notebook, we will explore a dataset of Disney movies and analyze what contributes to the success of Disney movies.

```
# Import pandas library

import pandas as pd
# Read the file into gross
gross = pd.read_csv("datasets/disney_movies_total_gross.csv",parse_dates=[“release_date"])
```



| movie_title | release_date | genre | mpaa_rating | total_gross | inflation_adjusted_gross | |
|---|---|---|---|---|---|---|
| 0 | Snow White and the Seven Dwarfs | 1937-12-21 | Musical | G | 184925485 | 522895325 |
| 1 | Pinocchio | 1940-02-09 | Adventure | G | 84300000 | 218822905 |
| 2 | Fantasia | 1940-11-13 | Musical | G | 83320000 | 218709080 |
| 3 | Song of the South | 1946-11-12 | Adventure | G | 65000000 | 107851057 |
| 4 | Cinderella | 1950-02-15 | Drama | G | 85000000 | 920608730 |

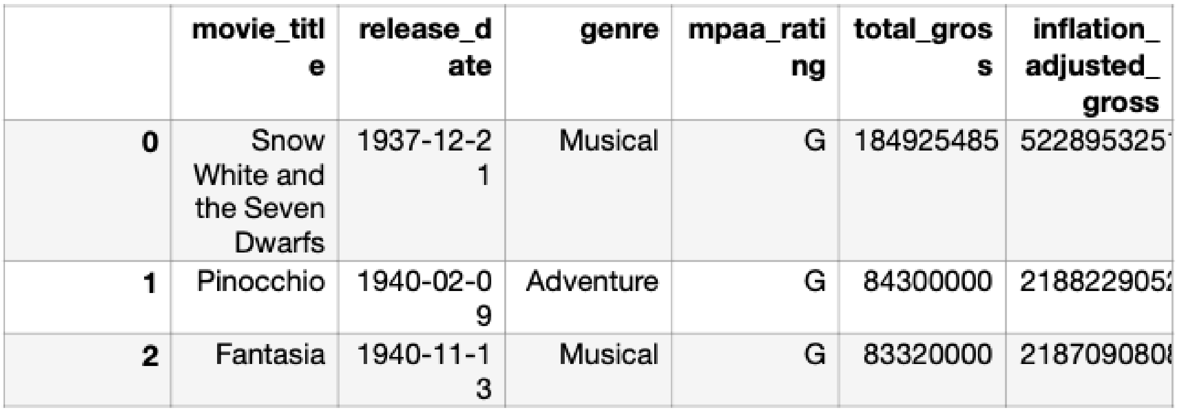
### 3.2 Data filtering

**-** Data filtering is the process of choosing a smaller part of your data set and using that subset for viewing or analysis. This **function** is useful when you want to focus only on specific information in a large dataset or table. **Filtering** doesn't remove or modify **data**; it just changes which records appear on your screen.

- Let's start by exploring the data. We will check which are the 10 Disney movies that have earned the most at the box office. We can do this by sorting movies by their inflation-adjusted gross (we will call it adjusted gross from this point onward).

In this step, the parameters chosen for visualization and other data science techniques are sorted and arranged.

```
# Sort data by the adjusted gross in descending order
inflation_adjusted_gross=gross.sort_values(by="inflation_adjusted_gross", ascending=False)

# Display the top 10 movies
inflation_adjusted_gross.head(10)
```



| | movie_title | release_date | genre | mpaa_rating | total_gross | inflation_adjusted_gross |
|---|---|---|---|---|---|---|
| **0** | Snow White and the Seven Dwarfs | 1937-12-21 | Musical | G | 184925485 | 522895325 |
| **1** | Pinocchio | 1940-02-09 | Adventure | G | 84300000 | 218822905 |
| **2** | Fantasia | 1940-11-13 | Musical | G | 83320000 | 218709080 |

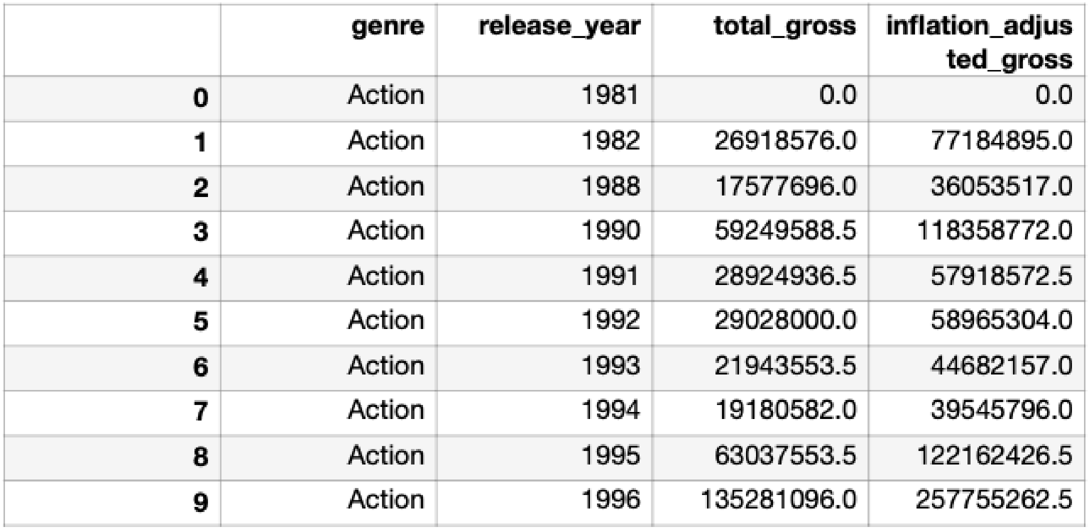
### 3.4 Data visualization

- **Data visualization** refers to the techniques used to communicate **data** or information by encoding it as visual objects (e.g., points, lines or bars) contained in graphics. The goal is to communicate information clearly and efficiently to users. It is one of the steps in **data analysis** or **data science**.

- From the top 10 movies above, it seems that some genres are more popular than others. So, we will check which genres are growing stronger in popularity. To do this, we will group movies by genre and then by year to see the adjusted gross of each genre in each year.

- **For our project, after filtering out the required parameters, we use Seaborn library with Python.**

```
# Extract year from release_date and store it in a new column
gross['release_year'] = pd.DatetimeIndex(gross[release_date]).year

# Compute mean of adjusted gross per genre and per year
group = gross.groupby(["genre","release_year"]).mean()

# Convert the GroupBy object to a DataFrame
genre_yearly = group.reset_index()

# Inspect genre_yearly
genre_yearly.head(10)
```

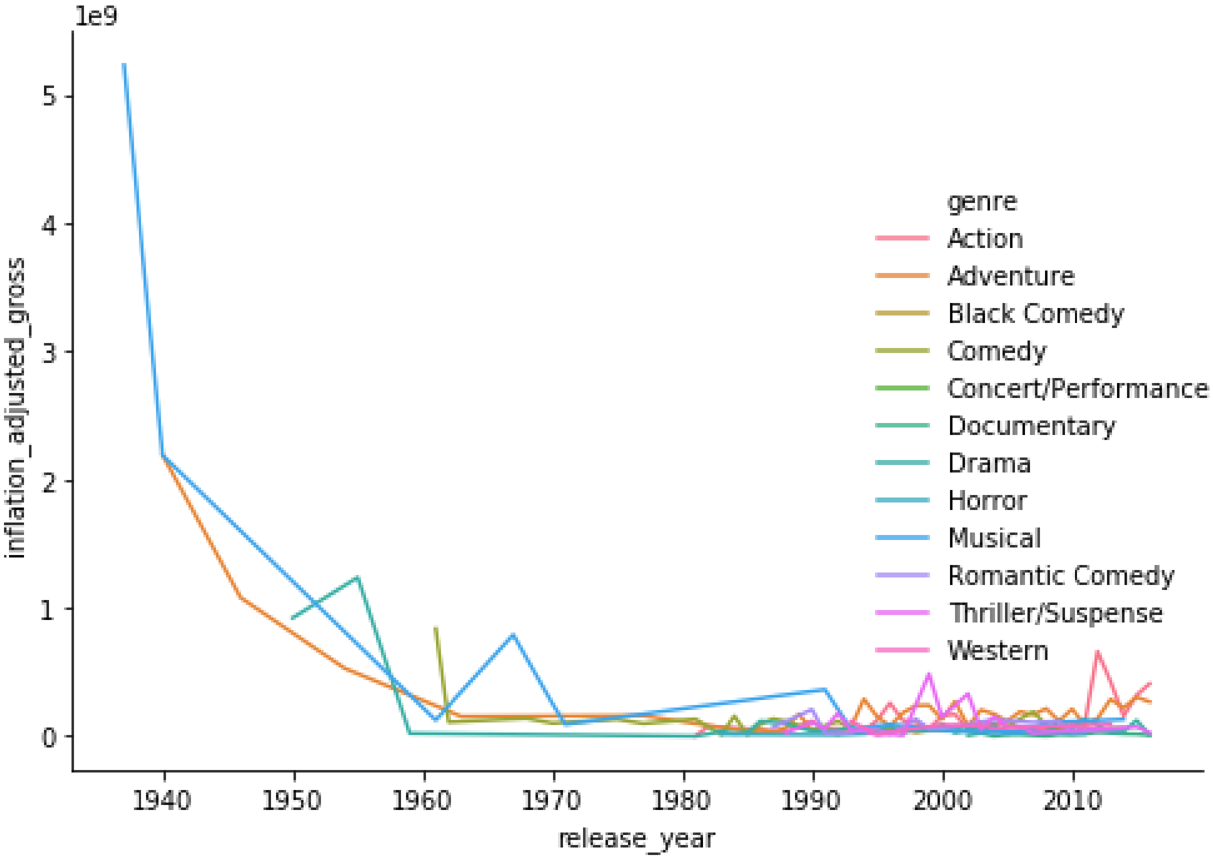


| | genre | release_year | total_gross | inflation_adjusted_gross |
|---|---|---|---|---|
| **0** | Action | 1981 | 0.0 | 0.0 |
| **1** | Action | 1982 | 26918576.0 | 77184895.0 |
| **2** | Action | 1988 | 17577696.0 | 36053517.0 |
| **3** | Action | 1990 | 59249588.5 | 118358772.0 |
| **4** | Action | 1991 | 28924936.5 | 57918572.5 |
| **5** | Action | 1992 | 29028000.0 | 58965304.0 |
| **6** | Action | 1993 | 21943553.5 | 44682157.0 |
| **7** | Action | 1994 | 19180582.0 | 39545796.0 |
| **8** | Action | 1995 | 63037553.5 | 122162426.5 |
| **9** | Action | 1996 | 135281096.0 | 257755262.5 |

- We will make a plot out of these means of groups to better see how box office revenues have changed over time**.**

```
# Import seaborn library
import seaborn as sns

# Plot the data
sns.relplot(x="release_year",
y="inflation_adjusted_gross",kind="line",hue="genre",data=genre_yearly)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f0aec2ea320>
```



## 3.5 Linear Regression

- line plot supports our belief that some genres are growing faster in popularity than others. For Disney movies, Action and Adventure genres are growing the fastest. Next, we will build a linear regression model to understand the relationship between genre and box office gross.
- Since linear regression requires numerical variables and the genre variable is a categorical variable, we'll use a technique called one-hot encoding to convert the categorical variables to numerical. This technique transforms each category value into a new column and assigns a 1 or 0 to the column.
- For this dataset, there will be 11 dummy variables, one for each genre except the action genre which we will use as a baseline. For example, if a movie is an adventure movie, like The Lion King, the adventure variable will be 1 and other dummy variables will be 0. Since the action genre is our baseline, if a movie is an action movie, such as The Avengers, all dummy variables will be 0.

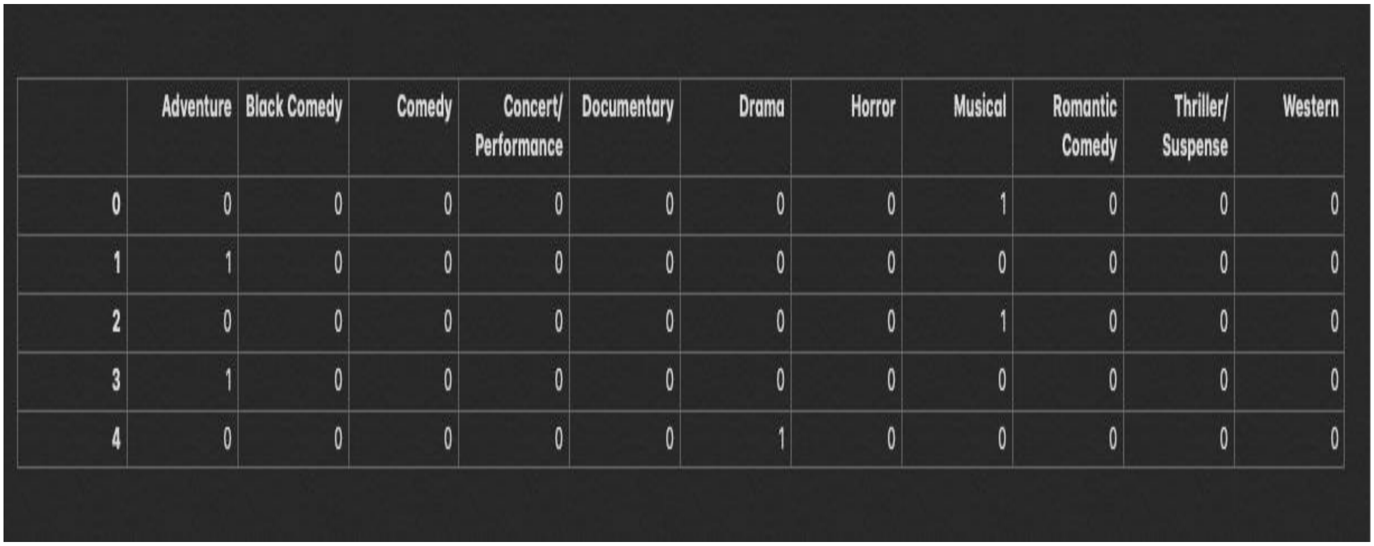
  **Computers are brilliant when dealing with numbers. So, we must somehow convert our input data (in whichever sequential format it be) to numbers.**

### ONE-HOT ENCODING-

- One-hot encoding is essentially the representation of categorical variables as binary vectors. These categorical values are first mapped to integer values. Each integer value is then represented as a binary vector that is all 0s (*except*the index of the integer which is marked as 1).
- As in our project, we are using the “movie_genre” and “income”. We assign the integer value of “movie_genre” to be 0 and income to be 1. Now we can create a binary vector array that can represent each integer value. Since there are 2 possible values, the vector will have 2 elements. The rule to make such a binary vector is simple.

- So after **one- hot encoding it becomes;**

```
# Convert genre variable to dummy variables
genre_dummies =pd.get_dummies(data=gross["genre”],drop_first=True)

# Inspect genre_dummies
genre_dummies.head()
```



| | Adventure | Black Comedy | Comedy | Concert/ Performance | Documentary | Drama | Horror | Musical | Romantic Comedy | Thriller/ Suspense | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

## The Genre effect:

- From the regression model, we can check the effect of each genre by looking at its coefficient given in units of box office gross dollars. We will focus on the impact of action and adventure genres here. (Note that the intercept and the first coefficient values represent the effect of action and adventure genres respectively). We expect that movies like the Lion King or Star Wars would perform better for box office.

```
# Import LinearRegression
from sklearn.linear_model import LinearRegression

# Build a linear regression model
regr = LinearRegression()

# Fit regr to the dataset
regr.fit(genre_dummies,inflation_adjusted_gross)

# Get estimated intercept and coefficient values
action =  regr.intercept_
adventure = regr.coef_[[0]][0]
```

```
# Inspect the estimated intercept and coefficient values
print((action, adventure))
```

**Output:**

```
(102921757.36842033, 87475654.70909958)
```

# Confidence intervals for regression parameters (i)

- The 95% confidence intervals for the intercept *a* and coefficient *bi* means that the intervals have a probability of 95% to contain the true value *a* and coefficient *bi* respectively. If there is a significant relationship between a given genre and the adjusted gross, the confidence interval of its coefficient should exclude 0.
- We will calculate the confidence intervals using the pairs bootstrap method.

```
# Import a module
import numpy as np

# Create an array of indices to sample from
inds = np.arange(len(gross[['genre']]))

# Initialize 500 replicate arrays
size = 500
bs_action_reps =  np.empty(size)
bs_adventure_reps = np.empty(size)
```

## Confidence intervals for regression parameters (ii)

- After the initialization, we will perform pair bootstrap estimates for the regression parameters. Note that we will draw a sample from a set of (genre, adjusted gross) data where the genre is the original genre variable. We will perform one-hot encoding after that.

```
# Generate replicates
for i in range(size):

    # Resample the indices
    bs_inds = np.random.choice(inds,size=len(inds))

    # Get the sampled genre and sampled adjusted gross
    bs_genre = gross['genre'][bs_inds]
    bs_gross = gross["inflation_adjusted_gross"][bs_inds]

    # Convert sampled genre to dummy variables
    bs_dummies = pd.get_dummies(data=bs_genre)

    # Build and fit a regression model
    regr = LinearRegression().fit(bs_dummies, bs_gross)

    # Compute replicates of estimated intercept and coefficient
    bs_action_reps[i] = regr.intercept_
    bs_adventure_reps[i] = regr.coef_[[0]][0]
```

## Confidence intervals for regression parameters (iii)

- Finally, we compute 95% confidence intervals for the intercept and coefficient and examine if they exclude 0. If one of them (or both) does, then it is unlikely that the value is 0 and we can conclude that there is a significant relationship between that genre and the adjusted gross.

```
  # Compute 95% confidence intervals for intercept and coefficient values
  confidence_interval_action = np.percentile(bs_action_reps,[2.5,97.5])
  confidence_interval_adventure = np.percentile(bs_adventure_reps,[2.5,97.5])

 # Inspect the confidence intervals
 print(confidence_interval_action)
 print(confidence_interval_adventure)
```

**o/p-**

```
[7.04284578e+07 2.23480716e+08]

[-1.14853915e+08  8.54028713e+07]
```

### 3.6 Future insights

- The confidence intervals from the bootstrap method for the intercept and coefficient do not contain the value zero, as we have already seen that lower and upper bounds of both confidence intervals are positive. These tell us that it is likely that the adjusted gross is significantly correlated with the action and adventure genres.

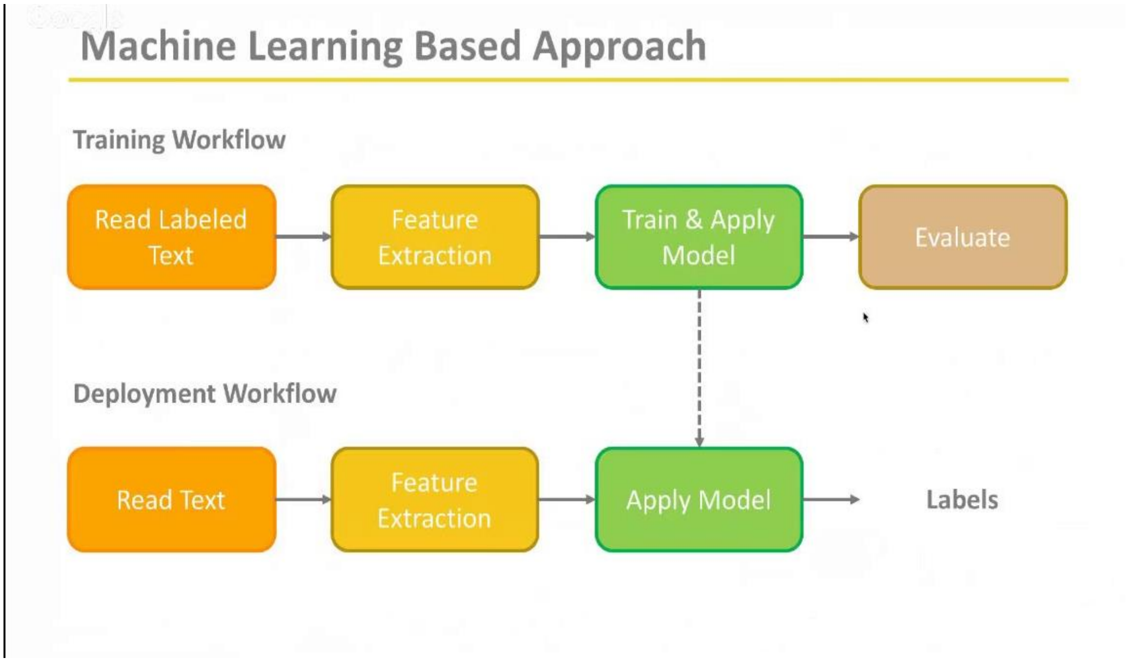
```
# should Disney studios make more action and adventure movies?

more_action_adventure_movies = more_action_adventure_movies = True
```

## CONCLUSION AND FUTURE SCOPE

From the results of the bootstrap analysis and the trend plot we have done earlier, we could say that Disney movies with plots that fit into the action and adventure genre, according to our data, tend to do better in terms of adjusted gross than other genres. With the line up of adventure and action genre movies, it’s possible that Disney will beat its own record and bring more profits along with consumer satisfaction.

# SENTIMENT ANALYSIS

- **Sentiment analysis** is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.



Machine Learning Based Approach

Training Workflow

Read Labeled Text → Feature Extraction → Train & Apply Model → Evaluate

Deployment Workflow

Read Text → Feature Extraction → Apply Model → Labels

- Movie reviews are an important way to gauge the performance of a movie. While providing a numerical/stars rating to a movie tells us about the success or failure of a movie quantitatively, a collection of movie reviews is what gives us a deeper qualitative insight on different aspects of the movie. A textual movie review tells us about the strong and weak points of the movie and deeper analysis of a movie review can tell us if the movie in general meets the expectations of the reviewer.

- Sentiment Analysis is a major subject in machine learning which aims to extract subjective information from the textual reviews. The field of sentiment of analysis is closely tied to natural language processing and text mining. It can be used to determine the attitude of the reviewer with respect to various topics or the overall polarity of review. Using sentiment analysis, we can find the state of mind of the reviewer while providing the review and understand if the person was "happy", "sad", "angry" and so on.

- In this project we aim to use Sentiment Analysis on a set of movie reviews given by reviewers and try to understand what their overall reaction to the movie was, i.e. if they liked the movie or they hated it. We aim to utilize the relationships of the words in the review to predict the overall polarity of the review.

- Few basic steps include:

  a) Data collection
  b) Pre-Processing
  c) wordcloud model building
  d) Labelling word vectors
  e) Testing
  f) Data visualization
  g) Result analysis

**Predictive Task:**

- The main aim of this project is to identify the underlying sentiment of a movie review on the basis of its textual information. In this project, we try to classify whether a person liked the movie or not based on the review they give for the movie. This is particularly useful in cases when the creator of a movie wants to measure its overall performance using reviews that critics and viewers are providing for the movie. The outcome of this project can also be used to create a recommender by providing recommendation of movies to viewers on the basis of their previous reviews. Another application of this project would be to find a group of viewers with similar movie tastes (likes or dislikes).

## 4.2 Implementation

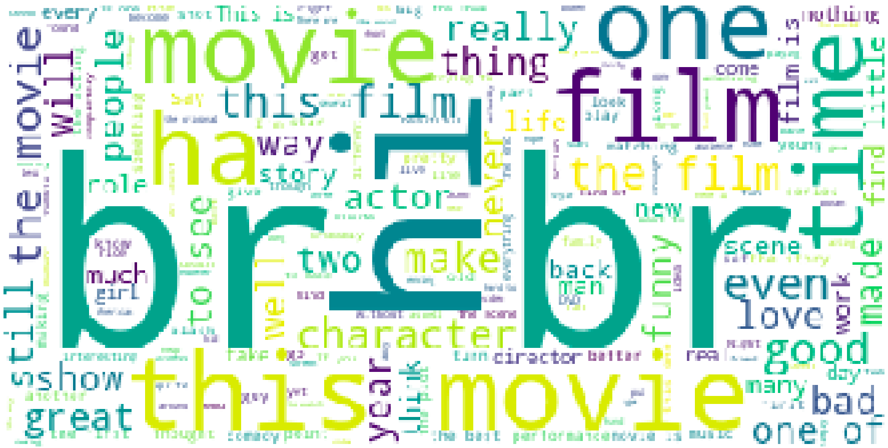
- As a part of this project, we aim to study several feature extraction techniques used in text mining e.g. keyword spotting, lexical affinity and statistical methods, and understand their relevance to our problem. In addition to feature extraction, we also look into different classification techniques and explore how well they perform for different kinds of feature representations. We finally draw a conclusion regarding which combination of feature representations and classification techniques are most accurate for the current predictive task.

- WordCloud is a technique to show which words are the most frequent among the given text. The first thing you may want to do before using any functions is check out the docstring of the function, and see all required and optional arguments.

- **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**.

- We start by forming a word cloud from the textual data present.

```
from wordcloud import WordCloud
from wordcloud import STOPWORDS
import matplotlib.pyplot as plt
fh=open("IMDB_sample.csv","r",encoding="utf8")
text=fh.read()
wordcloud=WordCloud(background_color="white",random_state=400,stopwords=set(STOPWORDS)).generate(text)
plt.imshow(wordcloud)
plt.axis("off")          #prevents the display of the x-axis and y-axis
plt.show()
```



- VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains. Each sentence's positivity, neutrality, negativity, and compound nature is scored by VADER and plotted for the entire debate.
- VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labelled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

- Additionally, it is then averaged across each answer by a candidate for a final debate score.

- The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive).
- positive sentiment : (compound score >= 0.05)  
  neutral sentiment : (compound score > -0.05) and (compound score < 0.05)  
  negative sentiment : (compound score <= -0.05)

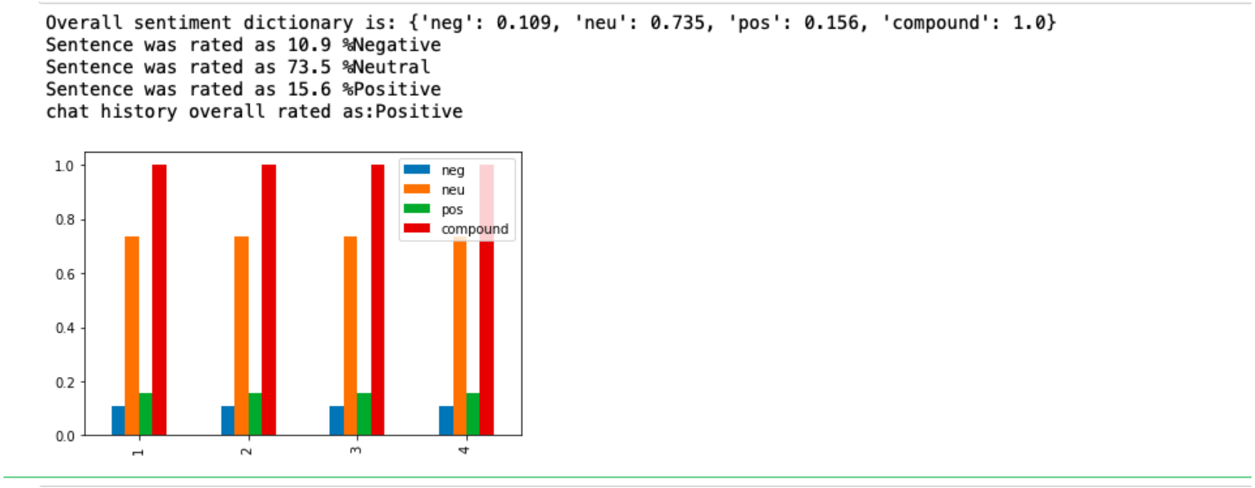
-

---

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

import pandas as pd

fh=open("IMDB_sample.csv","r",encoding="utf8")

text=fh.read()

sid_obj=SentimentIntensityAnalyzer()

sentiment_dict=sid_obj.polarity_scores(text)

print("Overall sentiment dictionary is:",sentiment_dict)

print("Sentence was rated as",sentiment_dict["neg"]*100,"%Negative")

print("Sentence was rated as",sentiment_dict["neu"]*100,"%Neutral")

print("Sentence was rated as",sentiment_dict["pos"]*100,"%Positive")

print("chat history overall rated as:",end="")
```

```
if sentiment_dict["compound"]>=0.05:

   print("Positive")

elif sentiment_dict["compound"]<=-0.05:

   print("Negative")

else:

   print("Neutral")

df=pd.DataFrame(sentiment_dict,columns=["neg","neu","pos","compound"],index=[1,2,3,4])

df.plot.bar()

plt.show()
```

**output:**



```
Overall sentiment dictionary is: {'neg': 0.109, 'neu': 0.735, 'pos': 0.156, 'compound': 1.0}
Sentence was rated as 10.9 %Negative
Sentence was rated as 73.5 %Neutral
Sentence was rated as 15.6 %Positive
chat history overall rated as:Positive
```

### 4.3 Inference drawn

- In above program, we tried to find the percentage of positive, negative and neutral reviews of movies. Sentiment analysis can be done by using various other techniques such as, Naïve Bayes, Random forest too.
- It is observed that consumers referring to IMDb, have an overall positive opinion about the service given by it. This gives us insight on which group of audience to target more further on, how to improve the service to acquire more consumers etc.

# References:

1. https://www.coursera.org/projects/neural-network-tensorflow
2. https://www.youtube.com/watch?v=GBZTd6NXHrE
3. https://youtu.be/7zzi7SmkCXw
4. https://www.youtube.com/watch?v=Zt1sCHsCvNc
5. https://www.youtube.com/watch?v=0HPes-mYpII
6. https://www.youtube.com/watch?v=JqSTk79xQa0
7. https://www.kaggle.com/speckledpingu/sentiment-analysis-with-vader
8. https://www.researchgate.net/publication/321843804_Sentiment_Analysis_of_Movie_Reviews_using_Machine_Learning_Techniques
9. https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f
10. https://amueller.github.io/word_cloud/index.html
11. https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html
12. https://www.kdnuggets.com/2019/03/beginners-guide-linear-regression-python-scikit-learn.html
13. https://stackoverflow.com/questions/34007308/linear-regression-analysis-with-string-categorical-features-variables
14. https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/
15. https://realpython.com/python-web-scraping-practical-introduction/
16. https://www.geeksforgeeks.org/implementing-web-scraping-python-beautiful-soup/
17. https://programminghistorian.org/en/lessons/sentiment-analysis
18. https://towardsdatascience.com/challenges-in-sentiment-analysis-a-case-for-word-clouds-for-now-6e598def5794