King Saud University

College of Computer and Information Sciences

Department of Software Engineering

SWE 486 – Cloud Computing and Big Data

Sentiment Analysis

PHASE 3

#	NAME	ID
1	Lina Alkhodhiri	441201109
2	Noura Alsaif	439200832
3	Alanoud Alomar	441201113

GROUP #	4
SUPERVISOR	L. Hailah Almazrua

Submission Date: *August 14, 2022*

Table of figures

Table of figures

1. people's opinion on comedy movies

- ## Libraries used:

Pandas: Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays [1].

Matplotlib: Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy [2].

Seaborn: Is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions [3].

Collections: Collections in Python are containers that are used to store collections of data, the most commonly use Python collections module is **Counter** The Counter() function in collections module takes an iterable or a mapping as the argument and returns a Dictionary [4].

- ## Overview

Our first data visualization aims to test the first hypothesis mentioned in phase one report. "People tend to watch movies of comedy nature. If the hypothesis is approved, *Elcinema* will give an insight into the best comedy movies. On the contrary, if we cannot agree to the hypothesis, this may indicate that we must know what the site's pioneers want more". In the following sub-sections, we are going to illustrate our approach to proving the preceding hypothesis and explain our findings.

- ## Findings

To elaborate on the approach before discussing the findings, we will go through the steps to explain how we got to our final verdict. Our hypothesis essentially states that people tend to watch comedic movies. So, the first step was to create a subset of the dataset that consists of review containing the words 'كوميدي', 'ساخر', 'مضحك', and 'افيهات'. These words explain to us people's interest in comedy. And by creating the subset, we can explore only the reviews of interest.

```
df1 = data_df[data_df['clean text'].str.contains('كوميدي')]
```

Figure 1. Extracting reviews containing 'كوميدي'

```
df2 = data_df[data_df['clean text'].str.contains('ساخر')]
```

Figure 2. Extracting reviews containing 'ساخر'

```
df3 = data_df[data_df['clean text'].str.contains('مضحك')]
```

Figure 3. Extracting reviews containing 'مضحك'

```
df4 = data_df[data_df['clean text'].str.contains('افيهات')]
```

Figure 4. Extracting reviews containing 'افيهات'

After that, the four dataframes were merged to represent the final dataframe of interest.

```
#merge the four dataframes into one dataframe that is df_comedy
frames = [df1,df2,df3,df4]
df_comedy = pd.concat(frames)
```

Figure 5. Merging the four dataframes

Now we want to know how many of these reviews are positive, negative, and neutral based on the classification. We used a bar chart to represent this information in a readable manner. The following figure illustrates the bar chart's setup.

```
#bar chart
plt.figure(figsize=[10,9]);
order_by_count = df_comedy.classification.value_counts().index
sns.countplot(data = df_comedy, x='classification', order = order_by_count);
plt.title('People\'s opinion on movies', fontsize=12, weight='bold');
plt.xlabel('People\'s opinion',fontsize=10, weight='bold');
plt.ylabel('Frequency',fontsize=10, weight='bold');
```

Figure 6. Bar chart setup

The resultant bar chart with the x-axis representing the class and the y-axis representing its respective frequency displayed results aligning with our hypothesis. Essentially, approving the initial hypothesis we stated about people's positive impression of comedy movies.

As depicted in the bar chart, we can see that most reviews contain the words 'كوميدي', 'ساخر', 'مضحك', and 'افيهات', were positive reviews. Which means that when people write reviews about comedy films, they are often positive reviews. There are, of course, negative reviews, but they are less than positive reviews

Non-negative reviews that is positive and neutral reviews make up approximately 63.8% of the reviews. This is a good and somewhat acceptable percentage, so we can say that our hypothesis has been approved.

In the end, these conclusions can be useful to help *Elcinema.com* management in presenting the best comedy movies.

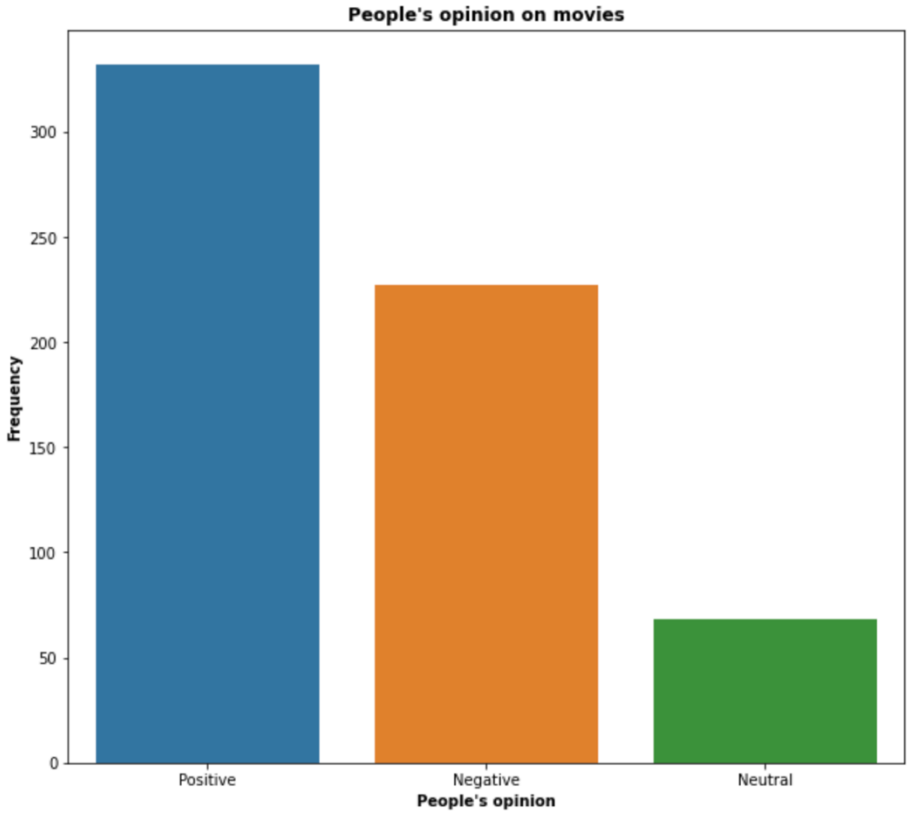

Figure 7. Resultant bar chart

To further visualize the results, another chart was created which is a pie chart. The following figure describes the setup.

```
#pie chart
plt.figure(figsize=[10,9]);
y = df_comedy.classification.value_counts()
myLabels = ["Positive","Neutral","Negative"]
plt.pie(y, labels= myLabels , autopct= '%1.1f%%', textprops={'fontsize':15, 'weight':'bold'})
plt.show()
```

Figure 8. Pie chart setup

The pie charts represent the classes in a pie shape with each section proportionate to its frequency. In addition, each section is labeled with its percentage. As shown, we can see that the largest percentage of reviews 53% belongs to the positive category and 10.8 % belongs to the neutral category, meaning that non-negative reviews amount to 63.8 %.

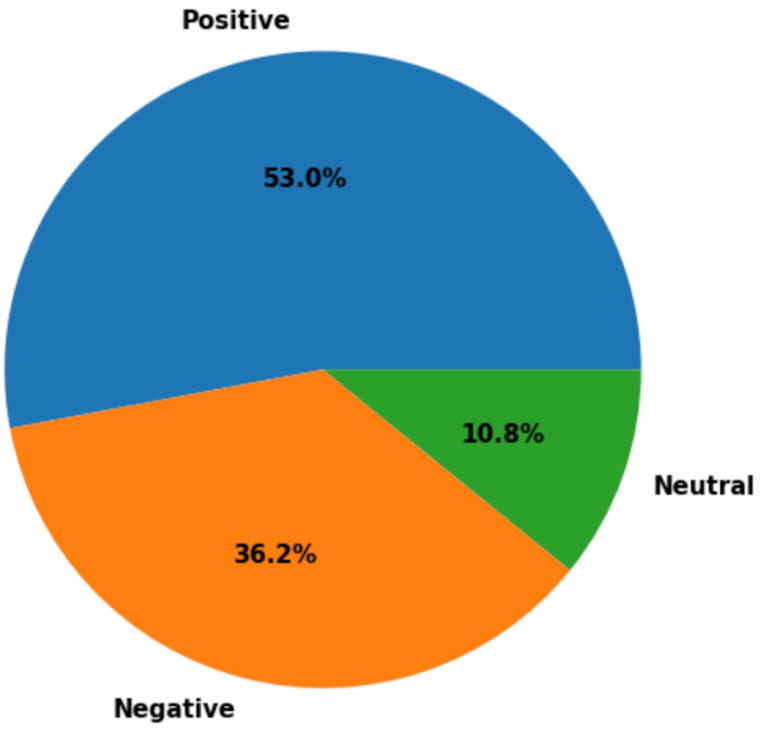

Figure 9. Resultant pie chart

- **Recommendations**

To advance our analysis in the project, it's recommended to :

- Collect more negative data sentiment to explore more about the comedy movise reviwes .
- Collect data from diffirent sources.
- use new tools and libraries.

2. People's opinion of long movies

- **Libraries used:**

Pandas: Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays [1].

Matplotlib: Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy [2].

Seaborn: Is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions [3].

Collections: Collections in Python are containers that are used to store collections of data, the most commonly use Python collections module is **Counter** The Counter() function in collections module takes an iterable or a mapping as the argument and returns a Dictionary [4].

- **Overview**

Our last data visualization aims to test the second hypothesis. “People tend to hate very long movies. If the hypothesis is approved, the *Elcinema* will help to focus more on the number of movies and display movies that attract viewers. On the contrary, if we cannot agree to the hypothesis, this means that *Elcinema* should display films of all kinds and may not pay attention to the duration of the movie”.

- **Findings**

To elaborate on the approach before discussing the findings, we will go through the steps to explain how we got to our final verdict. Our hypothesis essentially states that people tend to hate very long movies. So, the first step was to create a subset of the dataset that consists of review containing the words ‘وقت’ ,’مده’ ,’طويل’, and ‘طويل جدا’. These words show us the opinions of people. And by creating the subset, we can explore the validity of our hypothesis

```
df5 = data_df[data_df['clean text'].str.contains('وقت')]
```

Figure 10. Extracting reviews containing ‘وقت’

```
df6 = data_df[data_df['clean text'].str.contains('مده')]
```

Figure 11. Extracting reviews containing ‘مده’

```
df7 = data_df[data_df['clean text'].str.contains('طويـل')]
```

Figure 12. Extracting reviews containing 'طويل'

```
df8 = data_df[data_df['clean text'].str.contains('طويـل جدا')]
```

Figure 13. Extracting reviews containing 'طويل جدا'

After that, the four dataframes were merged to represent the final dataframe of interest.

```
#merge the four dataframes into one dataframe that is df_time
frame = [df5,df6,df7,df8]
df_time = pd.concat(frame)
```

Figure 14. Merging the four dataframes

Now we want to know how many of these reviews are positive, negative, and neutral based on the classification. We used a bar chart to represent this information in a readable manner. The following figure illustrates the bar chart's setup.

```
#bar chart
plt.figure(figsize=[10,9]);
orderbycount = df_time.classification.value_counts().index
sns.countplot(data = df_time, x='classification', order = orderbycount);
plt.title('People\'s opinion on time of movies', fontsize=12, weight='bold');
plt.xlabel('People\'s opinion',fontsize=10, weight='bold');
plt.ylabel('Frequency',fontsize=10, weight='bold');
```

Figure 15. Bar chart setup

The resultant bar chart with the x-axis representing the class and the y-axis representing its respective frequency displayed results and through which we will judge our hypothesis

As depicted in the bar chart, we can see that most reviews contain the words 'وقت', 'مده', 'طويل', and 'طويل جدا', were positive reviews. Which means that when people write reviews about the time of the movie, they are often positive reviews. There of course exist some negative reviews but compared to positive reviews are very few.

Negative reviews constitute about 26.8% of the reviews. This percentage is very few and unacceptable, so we can say that our hypothesis is incorrect.

In the end, these conclusions can be useful to help *Elcinema.com* Management, regardless of the time of the movies they show.

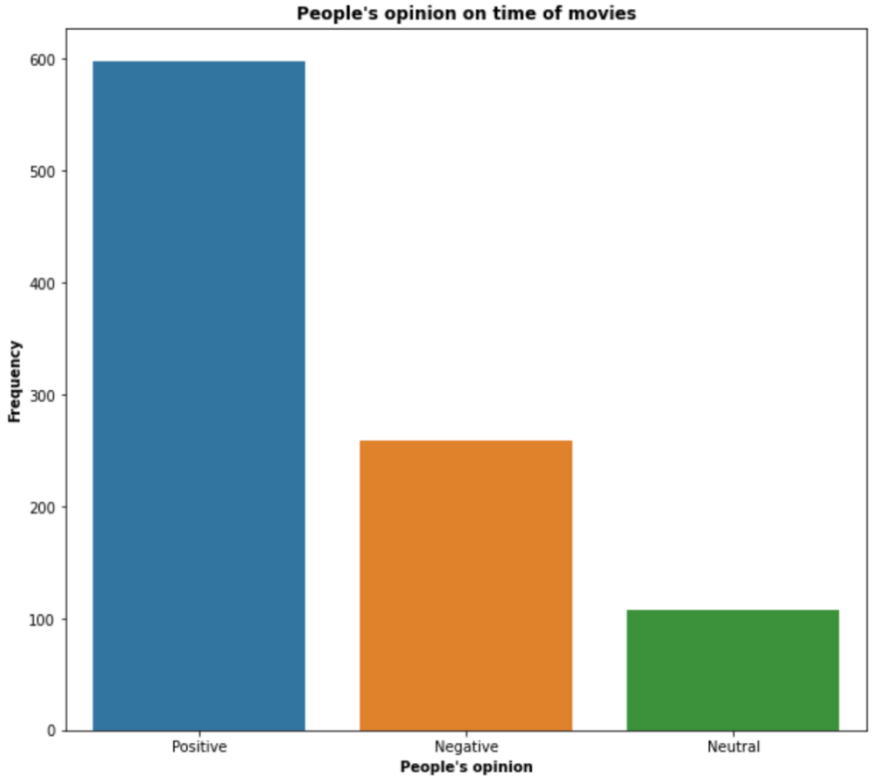

Figure 16. Resultant bar chart

To further visualize the results, another chart was created which is a pie chart. The following figure describes the setup.

```
#pie chart
plt.figure(figsize=[10,9]);
y = df_time.classification.value_counts()
myLabels = ["Positive","Negative","Neutral"]
plt.pie(y, labels= myLabels , autopct= '%1.1f%%', textprops={'fontsize':15, 'weight':'bold'})
plt.show()
```

Figure 17. Pie chart setup

The pie charts represent the classes in a pie shape with each section proportionate to its frequency. In addition, each section is labeled with its percentage. And as depicted, we can see that the largest percentage of reviews 62% This percentage contradicts our hypothesis.

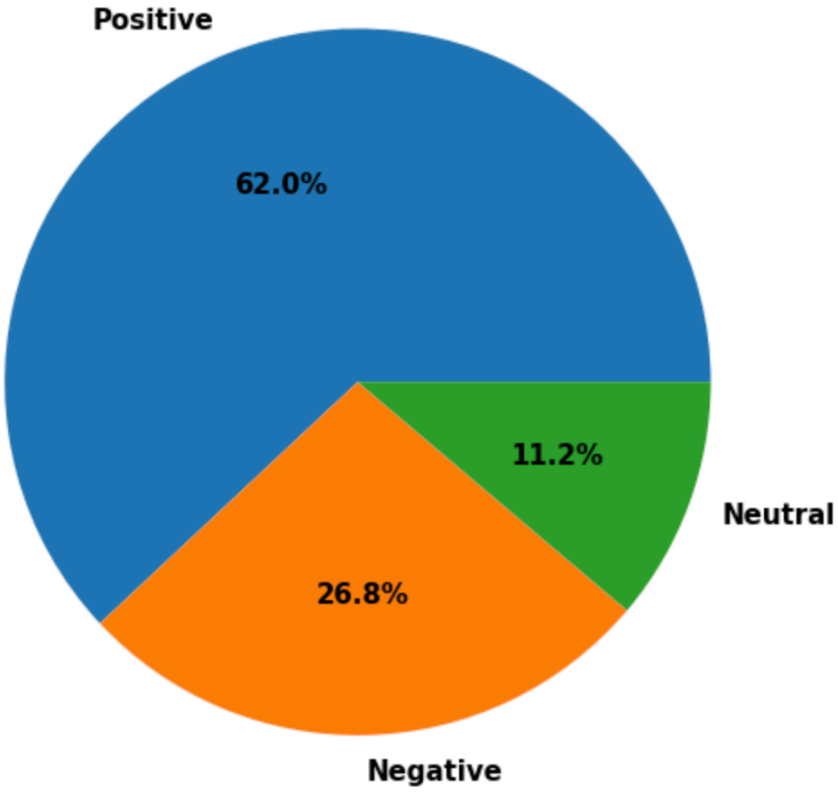

Figure 18. Resultant pie chart

- **Recommendations**

Based on our analysis, we recommend *Elcinema.com* to take under consideration the movie timing since most of the positive reviews was on this kind of movies. Also using new libraries and tools may improve our analysis.

References:

[1] https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/

[2] https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/

[3] https://www.w3schools.com/python/numpy/numpy_random_seaborn.asp

[4] https://stackabuse.com/introduction-to-pythons-collections-module/

Project file directory:

#	File Name	Description
1	SWE486_Phase1_Group#4	A PDF file containing the requirements for the first phase.
2	SWE486_Phase2_Group#4	A PDF file containing the requirements for the second phase.
3	SWE486_Phase3_Group#4	A PDF file containing the requirements for the third phase.
4	MovieCode.ipynb	Colab notebook, which contains all the codes used in the project except for visualizations code.
5	VisualizationsCode.ipynb	Colab notebook, which contains the visualizations code.
6	datasetMovie.xltx	Fail that contains the reviews after manual classification.