

Name: A.P.Raja Lakshmi

Collegecode: 9530

College name: st.Mother Theresa Engineering College

Team id:

Naan Mudhalvan id:au953021104037

Project Name: Image Recognition with IBM cloud

Problem Solution:

Consider incorporating sentiment analysis to generate captions that capture the emotions and mood of the images

Sentiment analysis is used ubiquitously to gain insights from text data. For example, by using sentiment analysis, companies can understand the voice of the customer or the market sentiment of the company. However, due to the lack of a standard infrastructure and standard libraries, many sentiment analysis projects remain at the proof-of-concept (POC) level and are never put into production.

With IBM Watson NLP, IBM introduced a common library for natural language processing, document understanding, translation, and trust. IBM Watson NLP brings everything under one umbrella for consistency and ease of development and deployment. This tutorial walks you through the process of using a pretrained model for sentiment analysis as well as fine-tuning a sentiment analysis model using the watson_nlp library.

The watson_nlp library is available on IBM Watson Studio as a runtime library so that you can directly use it for model training, evaluation, and prediction. The following figure shows the IBM Watson NLP architecture

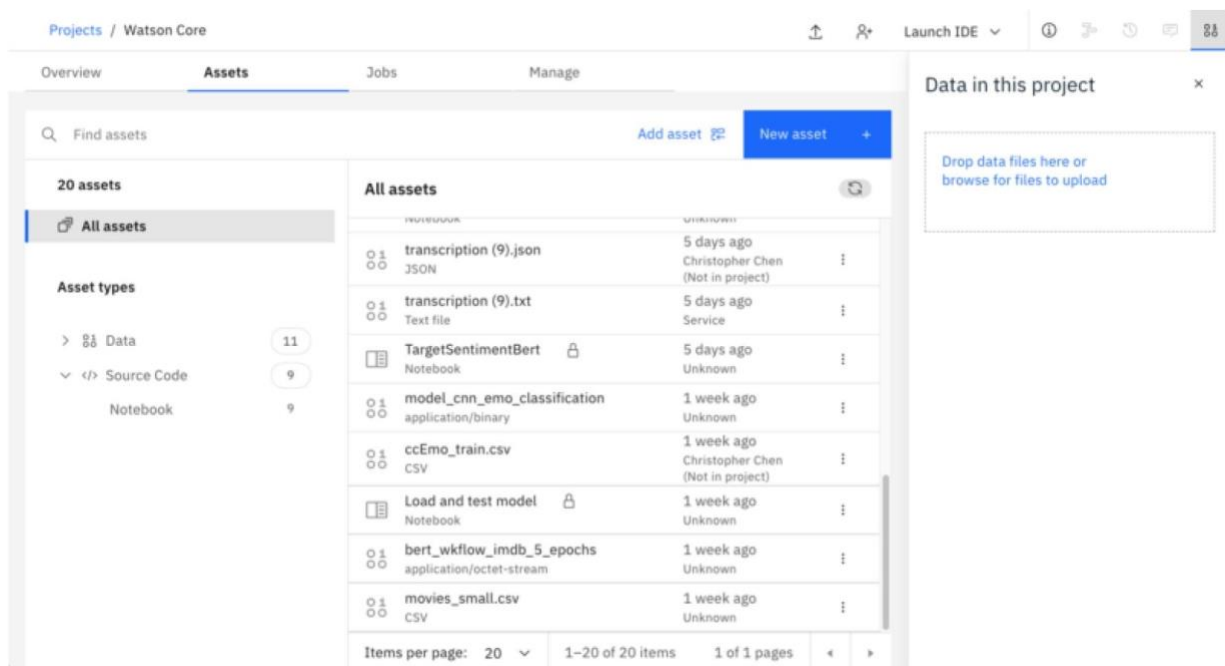
Steps

The steps in this tutorial use an example of IMDB movie reviews from Kaggle to walk you through the process.

Step 1. Collecting the data set

1.Download the IMDB movie reviews data set . This data set from Kaggle has been down-sampled and saved for you to reduce the model training time.

2.Upload the data set to your Watson Studio project by going to the Assets tab and then dropping the data files, as shown in the following figure.



After you add the data set to the project, you might have to reload the notebook. You have two options of accessing the data set from the Jupyter Notebook depending on the level of access that you have.

A. If you are a project administrator, then:

- i) You can just insert the project token as shown in the following image.
- ii) After inserting the project token, you can continue executing all of the cells in the notebook. This cell in particular loads your data set in the notebook.

```
In [5]: # load data set into a dataframe
buffer = project.get_file("movies_small.csv")
review_df = pd.read_csv(buffer)

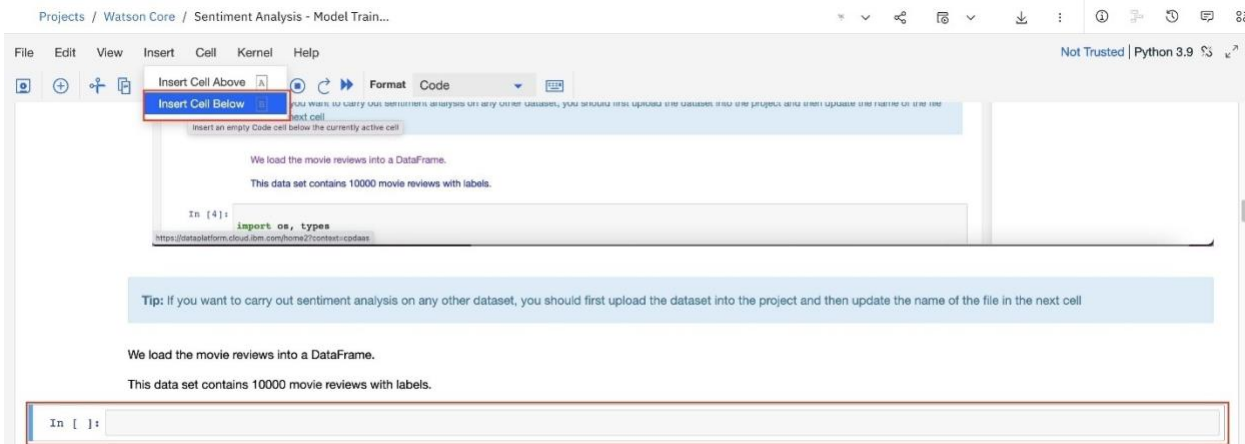
# preview the data set
review_df.head()
```

Out[5]:

	Unnamed: 0	text	label
0	0	I grew up (b. 1966) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The an...	0
1	1	When I put this movie in my DVD player, and sat down with a coke and some chips, I had some expectations. I was hoping that this movie would contain some of the strong points of the first movie: Awesome animation, good flying story, excellent voice cast, funny comedy and a kick-ass soundtrack. But, to my disappointment, not any of this is to be found in <i>Matrix: Min's Return</i> . Had I read some...	0
2	2	Why do people who do not know what a particular time in the past was like feel the need to try to define that time for others? Replace Woodstock with the Civil War and the Apollo moon-landing with the Titanic sinking and you've got as realistic a flick as this formulaic soap opera populated entirely by low-life trash. Is this what kids who were too young to be allowed to go to Woodstock and wh...	0
3	3	Even though I have great interest in Biblical movies, I was bored to death every minute of the movie. Everything is bad. The movie is too long, the acting is most of the time a joke and the script is horrible. I did not get the point in mixing the story about Abraham and Noah together. So if you value your time and sanity stay away from this horror.	0
4	4	In a die hard Dad's Army fan and nothing will ever change that. I got all the tapes, DVD's and audiobooks and every time I watch/listen to them its brand new. The film. The film is a re-run of certain episodes, Man and the hour, Enemy within the gates, Battle School and numerous others with a different edge. Introduction of a new General instead of Captain Square was a brilliant mov...	1

B. If you are not a Watson Studio project administrator, then you cannot create a project token.

- i) Create a new cell under Step 2 – Data Loading by clicking the Insert menu, and then selecting Insert Cell Below or the Esc+B keyboard shortcut. Highlight the code cell that is shown in the following image by clicking it.



- ii) Ensure you place the cursor below the commented line. Click the Find and add data icon (01/00) at the upper-right. Choose the Files tab, and pick the movies_small.csv file. Click Insert to code and choose pandas DataFrame. Rename the DataFrame from df_data_1 to review_df.

After you've added the data set to the project, you can access it from the Jupyter Notebook and read the .csv file into a pandas DataFrame.

2. Data Loading (movie reviews)

`<div class="alert alert-block danger-warning">`
`Insert dataset.</div>`

Before you can begin working on this notebook, you need to insert the data into the notebook.

1. Create a new cell by clicking on **Insert** menu and then selecting **Insert Cell Below** or **Esc+B** shortcut key
1. Highlight the code cell shown in the image below by clicking it.
1. Ensure you place the cursor below the commented line.
1. Click the Find and add data icon (01/00) on the top right corner.
1. Choose the Files tab, and pick the 'movies_small.csv' file.
1. Click Insert to code and choose pandas DataFrame.

In []:

Tip: If you want to carry out sentiment analysis on any other dataset, you should first upload the dataset into the project and then update the name of the file in the next cell

We load the movie reviews into a DataFrame.

This data set contains 10000 movie reviews with labels.

In [4]:

```
import os, types
```

https://dataplatform.cloud.ibm.com/home?context=cpdaas

Data

Files **Connections**

Upload one file at a time. All file types accepted. 5 GB max file size.

Drag and drop files here or upload.

movies_small.csv

Insert to code

pandas DataFrame

Credentials

	text	label
	I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The onl...	0
	When I put this movie in my DVD player, and sat down with a coke and some chips, I had some expectations. I was hoping that this movie would contain some of the strong-points of the first movie: Awsome animation, good flowing story, excellent voice cast, funny comedy and a kick-ass soundtrack. But, to my disappointment, not any of this is to be found in Atlantis: Milo's Return. Had I read some...	0
	Why do people who do not know what a particular time in the past was like feel the need to try to define that time for others? Replace Woodstock with the Civil War and the Apollo moon-landing with the Titanic sinking and you've got as realistic a flick as this formulaic soap opera populated entirely by low-life trash. Is this what kids who were too young to be allowed to go to Woodstock and wh...	0
	Even though I have great interest in Biblical movies, I was bored to death every minute of the movie. Everything is bad. The movie is too long, the acting is most of the time a Joke and the script is horrible. I did not get the point in mixing the story about Abraham and Noah together. So if you value your time and sanity stay away from this horror.	0
	Im a die hard Dads Army fan and nothing will ever change that. I got all the tapes, DVD's and audiobooks and every time i watch/listen to them its brand new. The film. The film is a re run of certain episodes, Man and the hour, Enemy within the gates, Battle School and numerous others with a different edge. Introduction of a new General instead of Captain Square was a brilliant mov...	1

Step 2. Data processing and exploratory data analysis

Step 2.1. Extract frequently mentioned movie aspects

In this step, you use the IBM Watson NLP pretrained syntax model to extract the most frequently mentioned aspects (nouns) in the movie reviews for each movie to understand what the audiences are talking about in their reviews.

Load the IBM Watson NLP syntax model.

```
Syntax_model = watson_nlp.load(watson_nlp.download('syntax_izumo_en_stock'))
```

The syntax model performs parts-of-speech tagging (for nouns, pronouns, verbs, and so on) so that you can filter out all of the nouns from the reviews. Instead of retrieving the nouns as they occur in the review, the lemmatized version of the nouns (for example, companies -> company) is stored. This lets you create more accurate statistics over all noun occurrences.

Use the helper functions to identify the lemmatized form of nouns in the review text.

```
Def extract_nouns(review_text):
```

```
    # converting text into lower case before processing
```

```
Review_text = review_text.lower()
```

```
Syntax_result = syntax_model.run(review_text, parsers=('lemma', 'part_of_speech'))
```

```
    # filter nouns
```

```
    Nouns = [token['lemma'] for token in syntax_result.to_dict()['tokens'] if ((token['part_of_speech'] == 'POS_PROPN' or token['part_of_speech'] == 'POS_NOUN') and len(token['lemma']) > 2)]
```

```
    # remove stopwords
```

```
    Nouns = [noun for noun in nouns if noun.upper() not in stopwords]
```

```
    Return list(nouns)
```

Extract the lemmatized nouns and show them with the review text in which they occurred.

```
Noun_df = create_noun_dataframe(review_df)
```

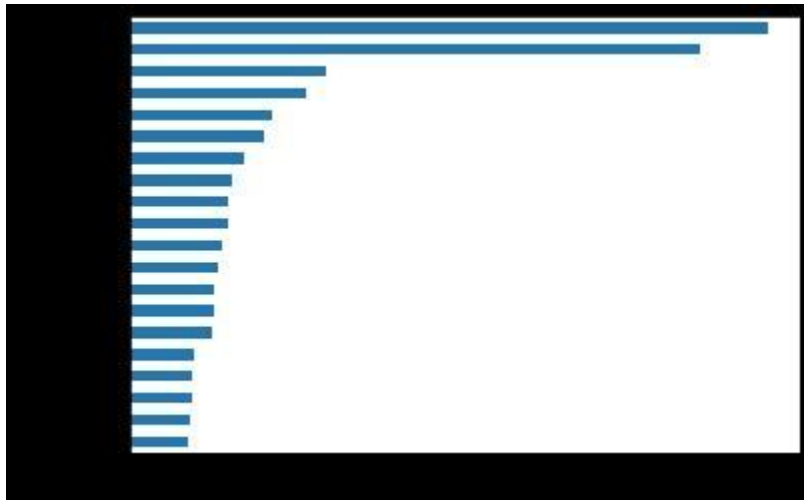
```
Sentiment_noun_df = review_df[['text', 'label']].merge(noun_df, how='left', left_index=True, right_index=True)
```

```
Sentiment_noun_df.head()
```

Use the explode function to transform the noun list to separate rows for each noun. That way, you can count the occurrences for each noun in a subsequent step.

```
Exp_nouns = sentiment_noun_df.explode('Nouns')
```

Plot the most frequent nouns as a bar chart.

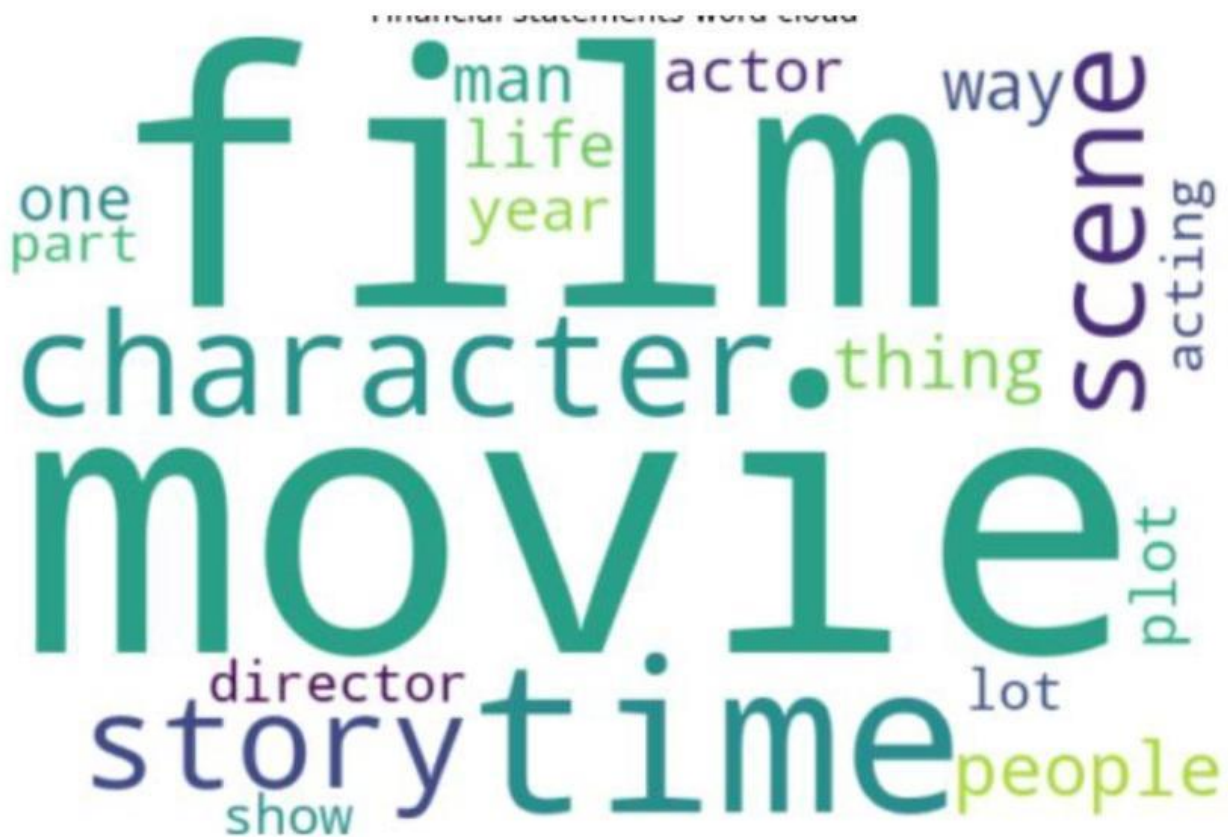


Create a word cloud for the most frequent nouns and show them.



This block extracts nouns from the movie reviews. The most frequently used nouns are typical aspects of a movie that review authors talk about.

text	label	Nouns
I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The onl...	0	[Thunderbird, mate, school, thunderbird, school, lunch, school, one, art, form, child, movie, glimpse, child, point, theme, tune, score, Thunderbird, Saturday, morning, television, channel, rerun, series, wife, director, chair, version, waste, film, rubbish, CGI, remake, marionette, Homo, error, judgment]
When I put this movie in my DVD player, and sat down with a coke and some chips, I had some expectations. I was hoping that this movie would contain some of the strong-points of the first movie: Awsome animation, good flowing story, excellent voice cast, funny comedy and a kick-ass soundtrack. But, to my disappointment, not any of this is to be found in Atlantis: Milo's Return. Had I read some...	0	[movie, player, coke, chip, expectation, movie, point, movie, animation, story, voice, cast, comedy, kick, soundtrack, disappointment, Return, review, paragraph, movie, point, scene, shock, Return, display, case, expectation, music, imitation, movie, voice, cast, exception, character, voice, Sweet, drawing, animation, sight, episode, story, time, episode, time, audience, movie, sequel, review,...]



Step 3. Model building

Step 3.1. Extract document and sentence sentiment

Are reviewers talking positively or negatively about the movies? Sentiment can be extracted for the complete review and for individual sentences. The sentiment extraction helpers can extract both sentiment levels.

Load the sentiment-aggregated_cnn-workflow_en_stock sentiment model for English.

```
Sentiment_model = watson_nlp.load(watson_nlp.download('sentiment-aggregated_cnn-workflow_en_stock'))
```

Extract the overall sentiment of the review and the sentiment for each sentence.

```
Def extract_sentiment(review_text):  
    # run the syntax model  
    # converting review text into lower case  
    Review_text = review_text.lower()  
    Syntax_result = syntax_model.run(review_text, parsers=('token', 'lemma', 'part_of_speech'))  
    # run the sentiment model on the result of the syntax analysis  
    Sentiment_result = sentiment_model.run(syntax_result, sentence_sentiment=True)  
  
    Document_sentiment = sentiment_result.to_dict()['label']  
  
    Sentence_sentiment = [(sm['span']['text'], sm['label']) for sm in  
        sentiment_result.to_dict()['sentiment_mentions']]  
  
    Return (document_sentiment, sentence
```

Extract the sentiment and display it with the review text.

	text	label	Document Sentiment
	I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The onl...	0	SENT_NEGATIVE
	When I put this movie in my DVD player, and sat down with a coke and some chips, I had some expectations. I was hoping that this movie would contain some of the strong-points of the first movie: Awsome animation, good flowing story, excellent voice cast, funny comedy and a kick-ass soundtrack. But, to my disappointment, not any of this is to be found in Atlantis: Milo's Return. Had I read some...	0	SENT_NEGATIVE
	Why do people who do not know what a particular time in the past was like feel the need to try to define that time for others? Replace Woodstock with the Civil War and the Apollo moon-landing with the Titanic sinking and you've got as realistic a flick as this formulaic soap opera populated entirely by low-life trash. Is this what kids who were too young to be allowed to go to Woodstock and wh...	0	SENT_NEGATIVE
	Even though I have great interest in Biblical movies, I was bored to death every minute of the movie. Everything is bad. The movie is too long, the acting is most of the time a Joke and the script is horrible. I did not get the point in mixing the story about Abraham and Noah together. So if you value your time and sanity stay away from this horror.	0	SENT_NEGATIVE
	Im a die hard Dads Army fan and nothing will ever change that. I got all the tapes, DVD's and audiobooks and every time i watch/listen to them its brand new. The film. The film is a re run of certain episodes, Man and the hour, Enemy within the gates, Battle School and numerous others with a different edge. Introduction of a new General instead of Captain Square was a brilliant mov...	1	SENT_POSITIVE

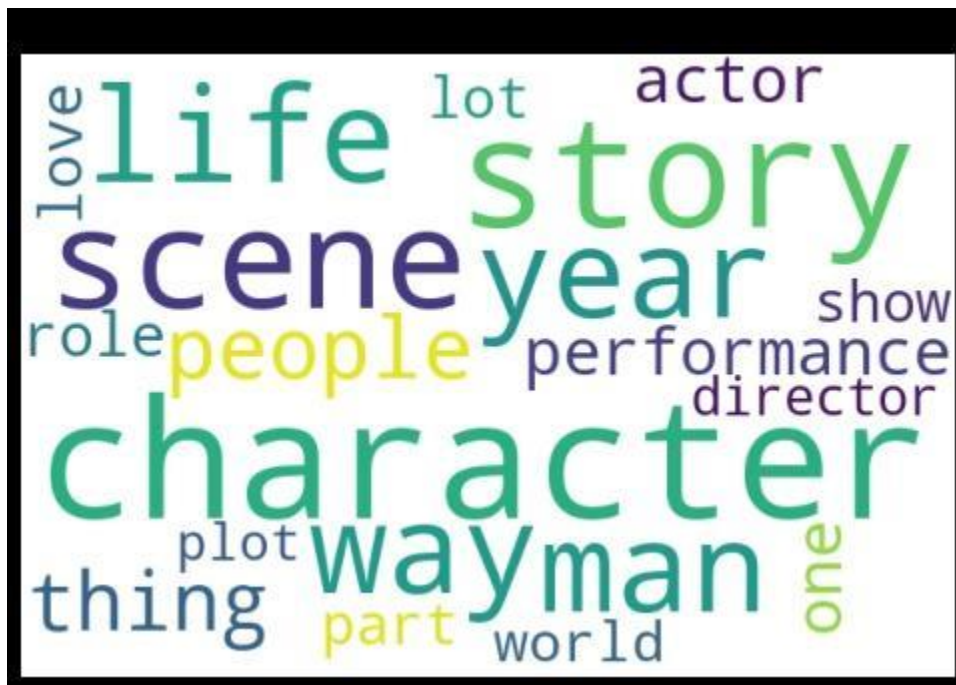
Step 3.2. Identify nouns that drive sentiment

You now identify the most frequently used nouns in sentences with positive or negative sentiment.

Extract nouns from the sentences/sentiment DataFrame.

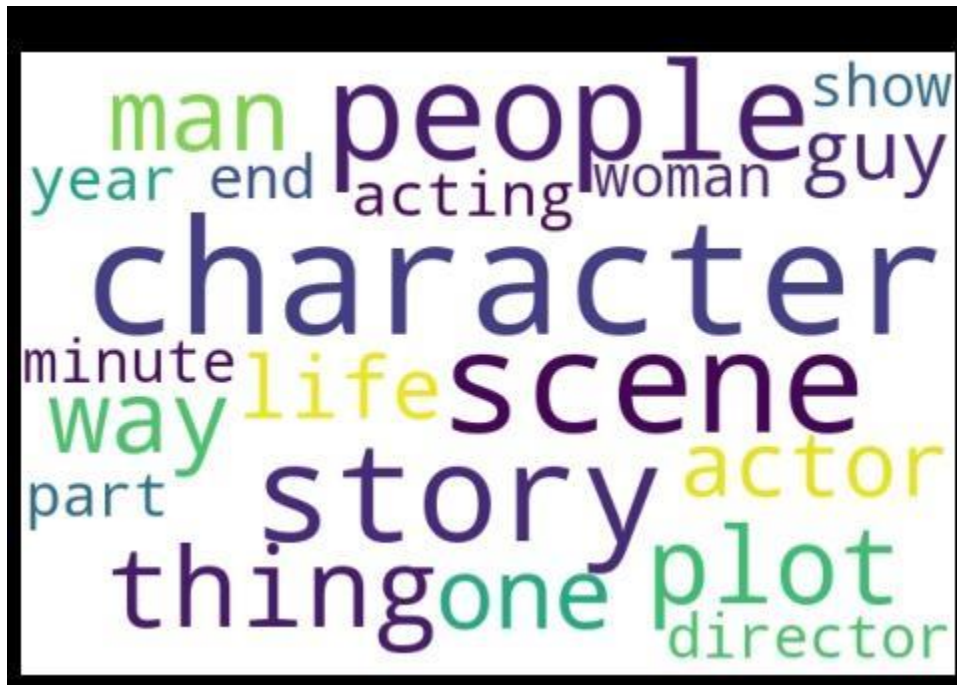
	Sentence	Sentiment	Nouns
	i grew up (b.	SENT_NEUTRAL	[]
	1965) watching and loving the thunderbirds.	SENT_POSITIVE	[thunderbird]
	all my mates at school watched.	SENT_POSITIVE	[mate, school]
	we played "thunderbirds" before school, during lunch and after school.	SENT_NEUTRAL	[thunderbird, school, lunch, school]
	we all wanted to be virgil or scott.	SENT_NEUTRAL	[]

Show the most frequent nouns in positive sentences.



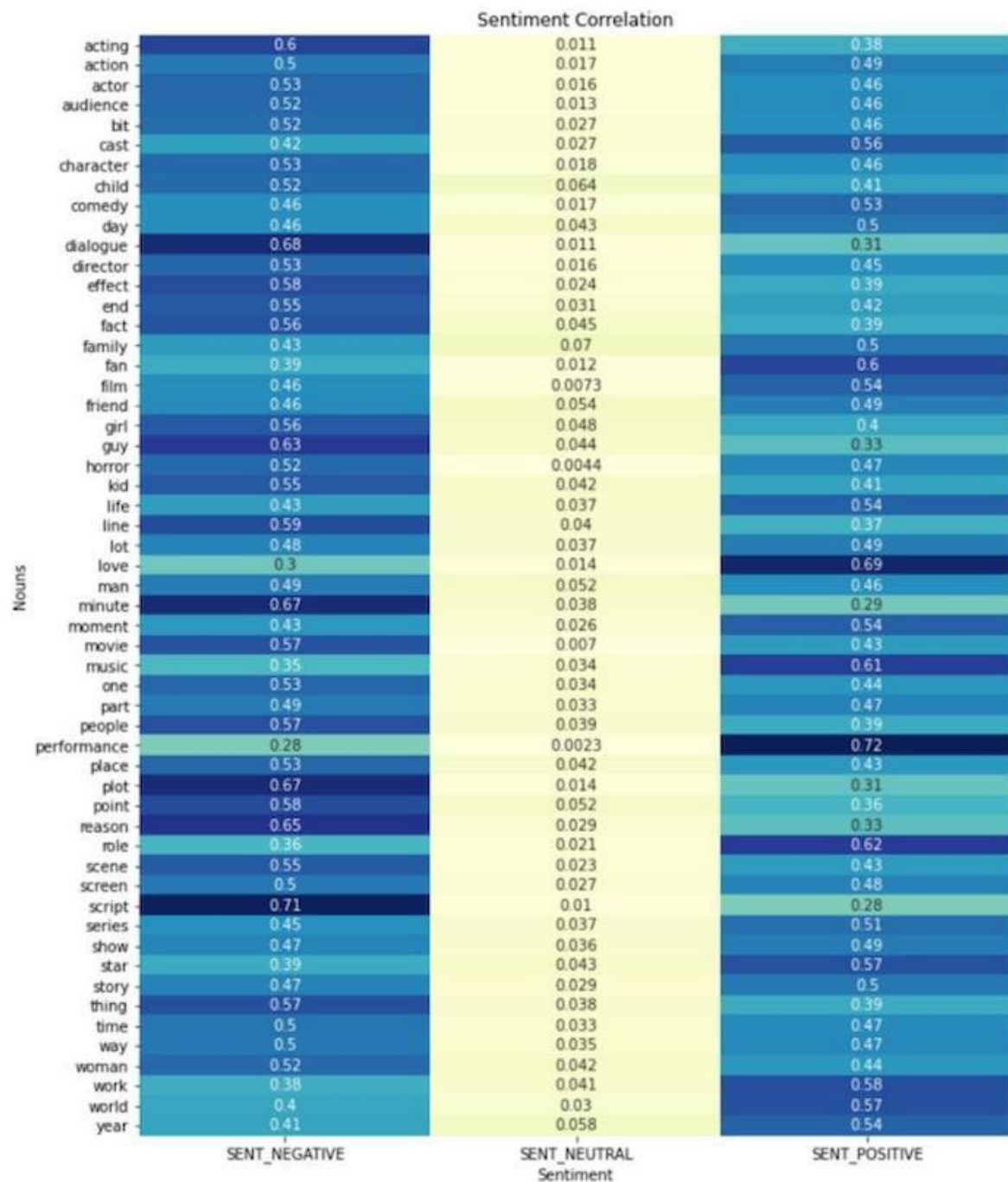
In this case, the word cloud shows that the most positive sentiments come from the reviews with words like film, movie, time, story, and character. So, the audience must love the film or movie when the story and character were good, and these were mentioned in the reviews.

Show the most frequent nouns in negative sentences.



In this case, the word cloud shows that the most negative sentiments come from reviews with words like time, character, scene, and director. So, the audience might not have liked movies with poor direction, character, and scenes in the movie.

Show the nouns that “drive” sentiment for each review. Create a cross tab between nouns and the resulting sentence sentiment and correlate them. The darker the cell, the more often a noun occurs in a sentence of a certain polarity.



You can see from the previous image that script is the most contributing factor to negative sentiments while performance and love are the most contributing factors to positive sentiments.

Step 3.3. Aspect-oriented sentiment analysis

Let's see what the sentiment for these aspects (script, performance, and love) looks like. For this, use the Watson NLP targeted sentiment model to extract the sentiment that is specific to the most frequently used nouns in the movie reviews.

Load the syntax model for English and the targeted BERT sentiment model.

```
Targeted_sentiment_model = watson_nlp.load(watson_nlp.download('sentiment-targeted_bert_multi_stock'))
```

Use a helper method to extract the sentiment for each target and the sentences containing the target as evidence.

```
Def extract_targeted_sentiment(review_text, target_words):  
    # run the syntax model  
  
    Syntax_result = syntax_model.run(review_text, parsers=('token', 'lemma', 'part_of_speech'))  
  
    # extract the spans for the target words  
  
    Targets = watson_nlp.toolkit.get_target_spans(review_text, target_words)  
  
    # run the sentiment model on the result of syntax and the target words  
  
    Sentiment_result = targeted_sentiment_model.run(syntax_result, targets)  
  
    # iterate over all target aspects  
  
    Target_sentiments = []  
  
    For idx, val in enumerate(target_words):  
  
        Sentiment_prediction = sentiment_result.to_dict()['sentiment_predictions'][idx]  
  
        Sentiment_mentions = None  
  
        If sentiment_prediction['sentiment_mentions'] is not None:  
  
            Sentiment_mentions = [(sm['span']['text'], sm['label']) for sm in  
            sentiment_prediction['sentiment_mentions']]  
  
        Target_sentiments.append(sentiment_prediction['label'])
```

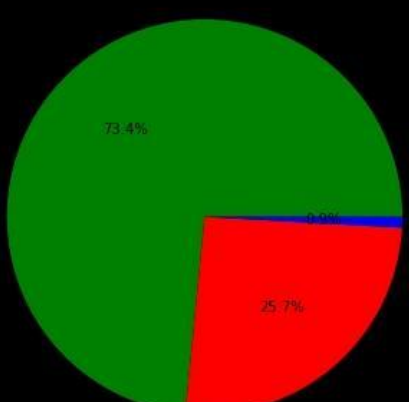
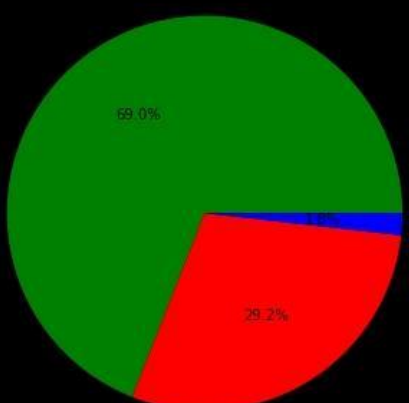
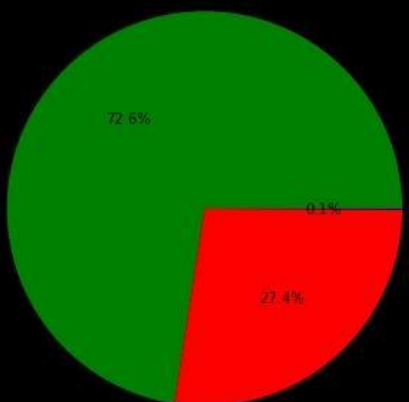
```
Target_sentiments.append(sentiment_mentions)
```

```
Return target_sentiments
```

Extract the sentiment specifically for script, performance, and love. Note that this cell runs for several minutes.

text	label	Sentiment 'performance'	Evidences 'performance'	Sentiment 'script'	Evidences 'script'	Sentiment 'love'	Evidences 'love'
I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The onl...	0	SENT_UNSET	None	SENT_UNSET	None	SENT_POSITIVE	[(I took my children to see the movie hoping they would get a glimpse of what I loved as a child., SENT_POSITIVE)]
When I put this movie in my DVD player, and sat down with a coke and some chips, I had some expectations. I was hoping that this movie would contain some of the strong-points of the first movie: Awsome animation, good flowing story, excellent voice cast, funny comedy and a kick-ass soundtrack. But, to my disappointment, not any of this is to be found in Atlantis: Milo's Return. Had I read some...	0	SENT_UNSET	None	SENT_UNSET	None	SENT_UNSET	None

Display the sentiment for selected aspects.



Step 4. Model evaluation

Apply data processing on the test data to make it compatible with the evaluate method in the Watson NLP library.

```
Def input_data_prep(df):  
    Df['weight'] = 1  
    Df.rename(columns={'label': 'labels'}, inplace=True)  
    Df = df[['text', 'weight', 'labels']]  
    Df['labels'] = df['labels'].replace({0: 'negative', 1: 'positive'})  
    Df['labels'] = df['labels'].apply(convertToList)  
    Display(df.head(5))  
    Return df
```

Evaluate the model by passing the test data and using the evaluate_quality method from the Watson NLP library.

```
Pre_eval_func = lambda x: syntax_model.run_batch(x)
```

```
Sentiment_model.find_label_func
```

```
Results = sentiment_model.evaluate_quality
```


results

```
{'negative': {'precision': 0.8587268208755496,
  'recall': 0.8907396391037081,
  'f1-score': 0.8744403348257738,
  'support': 5043},
'neutral': {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 0},
'positive': {'precision': 0.884784889821616,
  'recall': 0.8505144240468026,
  'f1-score': 0.8673112528286361,
  'support': 4957},
'accuracy': 0.8708,
'macro avg': {'precision': 0.5811705702323885,
  'recall': 0.5804180210501703,
  'f1-score': 0.58058386255147,
  'support': 10000},
'weighted avg': {'precision': 0.8716438056521146,
  'recall': 0.8708,
  'f1-score': 0.8709064488797925,
  'support': 10000}}
```

As you can see, the overall accuracy, precision and recall values are 0.87 each. This has been achieved by evaluating a pretrained model without training it on the IMDB Movie reviews data set

Git hub link :