WEM Groupe L
Manga Feelings

Denis Iglesias
Yves Kalberer
Thomas Zimmermann

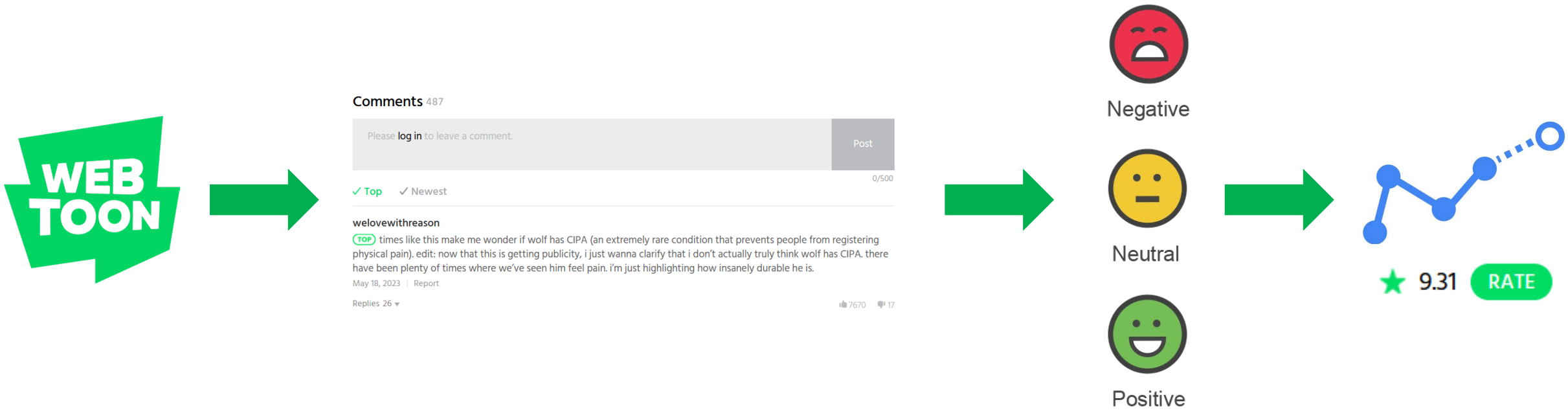
Vendredi 23 juin 2023

Table des matières

Introduction

Objectif

Scraping des commentaires du site webtoons pour faire du sentiment analysis et essayer de prédire la note attribuée la note du webtoon



Introduction

Données

- Source
  - https://www.webtoons.com/en/
- Webtoon
  - Bande dessinée coréenne (manwha) publiée en ligne. S'applique aujourd'hui a toutes les bande dessinée en ligne.
  - Hebdomadaire/saison
- TOS
  - Utilisation personnelle non commerciale
- Quantité de données
  - Un webtoon populaire par genre
  - 8 webtoons → 1472 épisode

série	épisodes
A Life Through Selfies	316
Goth Girl The Jock	32
How To Be A Dragon	36
My Husband Changes Every Night	5
Nerd And Jock	198
Power Pills	477
Seekers Log	168
Weakhero	240

Scraping

Technologies utilisées

- Site complexe
  - Javascript
  - Chargement dynamique du site

- Pas fonctionné
  - Scrapy

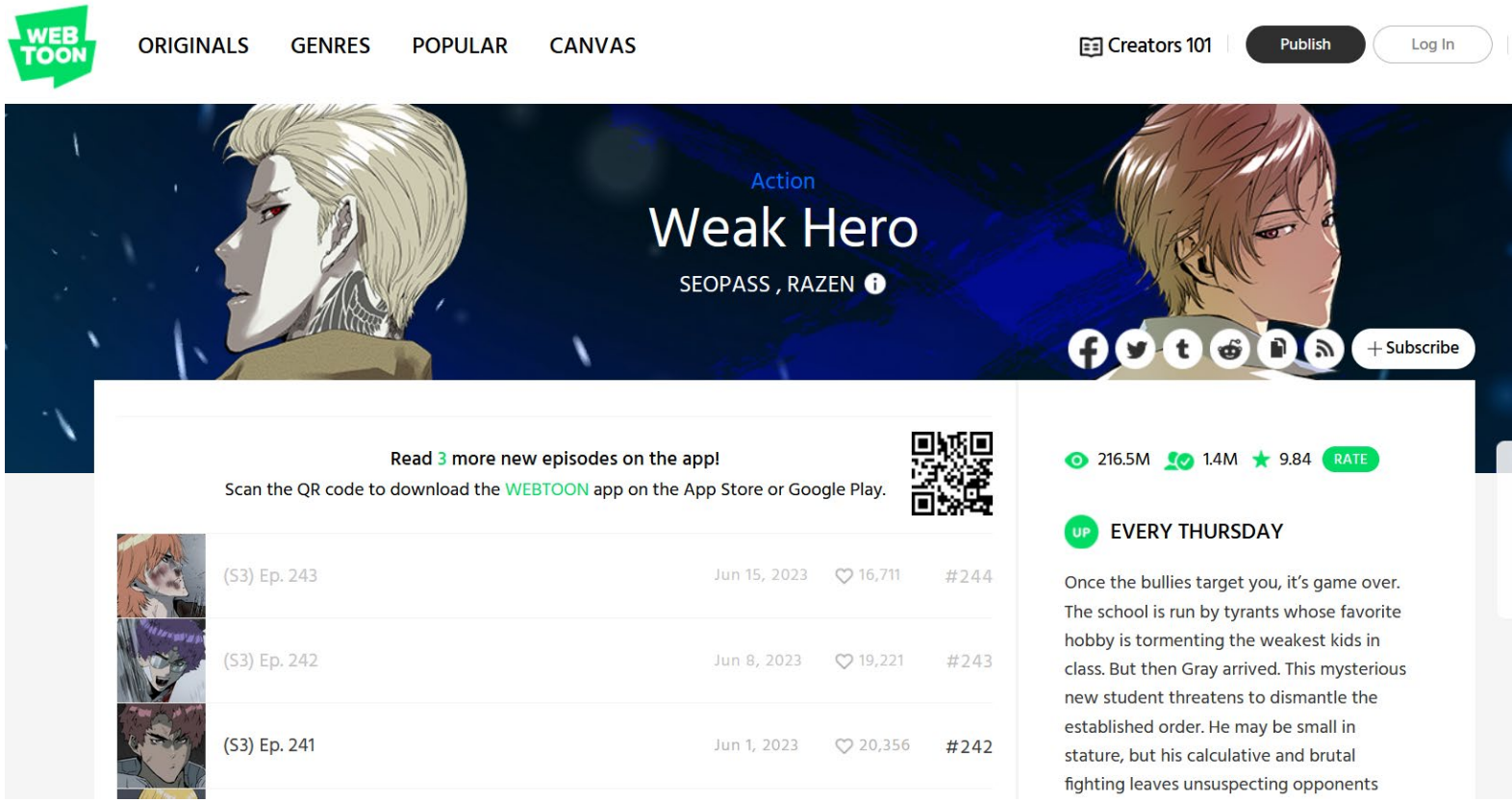
- Principales librairies utilisées

BeautifulSoup4

Scraping

Stratégie pour obtenir les informations des épisodes

1. Utilisation de Selenium
2. Renseigner url principale
3. Extraire
    1. Episode name
    2. Episode url
    3. Episode likes
    4. Episode Date
4. Page suivante

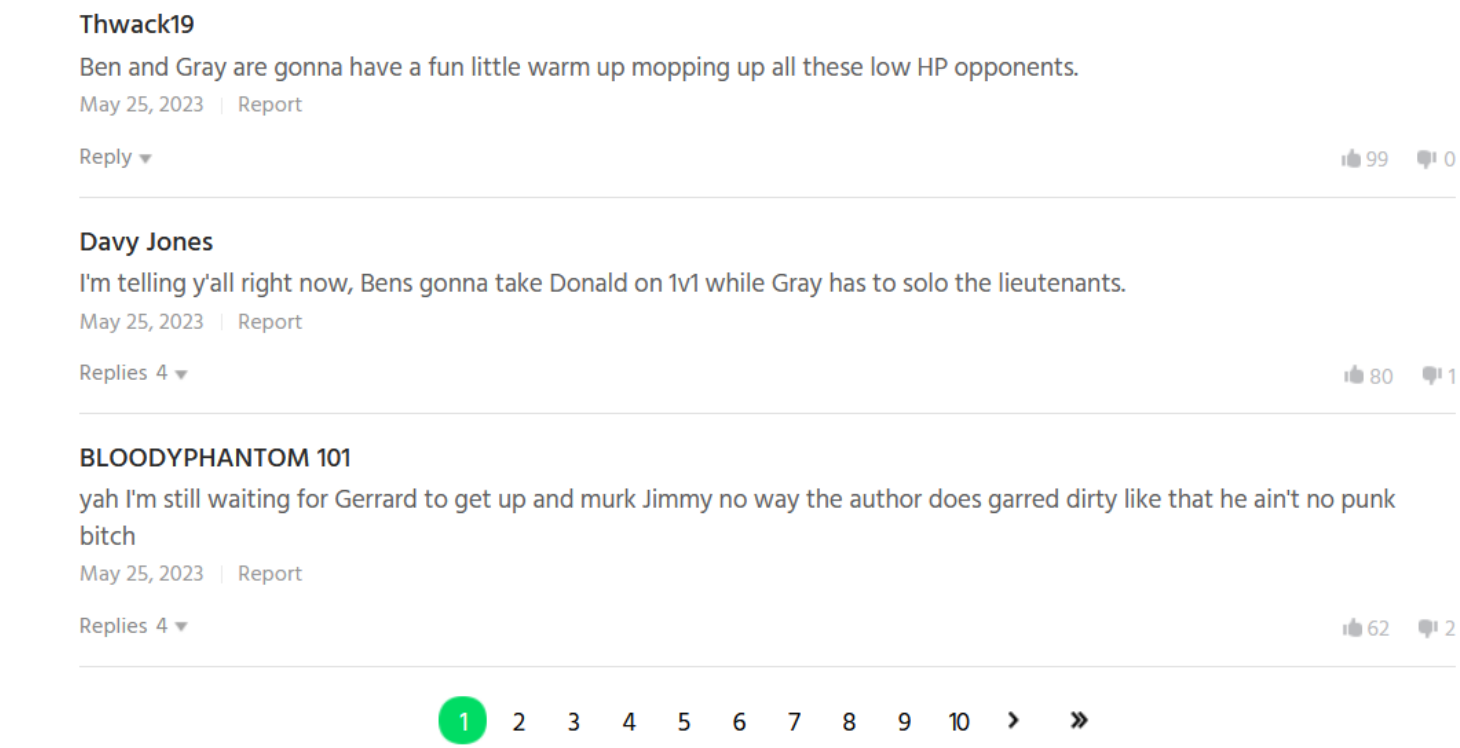


```python
def extract_episodes_info(driver, main_url, min_page, max_page):

    episodes_info = []

    for page_num in range(min_page, max_page + 1):
        if page_num == min_page:
            page_url = f"{main_url}&page={page_num}?lang=en"
        else:
            page_url = f"{main_url}&page={page_num}"

        driver.get(page_url)

        episode_items = driver.find_elements(By.CSS_SELECTOR, 'li._episodeItem')

        for episode_item in episode_items:
            episode_name = episode_item.find_element(By.CSS_SELECTOR, 'span.subj').text.strip()
            episode_url = episode_item.find_element(By.CSS_SELECTOR, 'a').get_attribute('href')
            likes_text = episode_item.find_element(By.CSS_SELECTOR, 'span.like_area').text.strip() # episode likes
            likes_text = likes_text.replace(",", "")  # Remove commas
            episode_likes = int(re.search(r'\d+', likes_text).group())
            episode_date = episode_item.find_element(By.CSS_SELECTOR, 'span.date').text.strip()

            episodes_info.append({'name': episode_name, 'url': episode_url,
                                  'likes': episode_likes, 'date': episode_date})
    return episodes_info
```

Scraping

Stratégie pour obtenir les commentaires

1. Aller dans url épisode
2. Descendre bas de page
3. Ouvrir tous les «reply»
4. Télécharger html (bs4)
5. Parse commentaire/reply
    1. Date
    2. User
    3. ID Commentaire
    4. Parent ID (pour reply)
    5. Like/dislike
6. Fermer tous les «reply»
7. Page commentaires suivante



```python
soup = BeautifulSoup(driver.page_source, 'html.parser')
comments = soup.select('.u_cbox_comment')

comment_data = []

for comment in comments:
    data_info = comment['data-info']
    data_info = data_info.replace("'", "").replace(" ", "")
    data_info_dict = dict(item.split(":") for item in data_info.split(",") if ":" in item)

    comment_id = data_info_dict.get('commentNo')
    reply_level = data_info_dict.get('replyLevel')
    parent_comment_no = data_info_dict.get('parentCommentNo')
    comment_text = comment.select_one('.u_cbox_contents').text
    comment_date = comment.select_one('.u_cbox_date')['data-value']
    comment_author = comment.select_one('.u_cbox_nick').text

    # Extract likes and dislikes
    likes = comment.select_one('.u_cbox_tool .u_cbox_cnt_recomm').text
                if comment.select_one('.u_cbox_tool .u_cbox_cnt_recomm') else "0"
    dislikes = comment.select_one('.u_cbox_tool .u_cbox_cnt_unrecomm').text
                if comment.select_one('.u_cbox_tool .u_cbox_cnt_unrecomm') else "0"

    comment_data.append([episode_url, episode_name, episode_likes, episode_date,\
                         comment_id, reply_level, parent_comment_no, comment_text,\
                          comment_date, comment_author, likes, dislikes])
```

Scraping

Résultat

- 1 csv par épisode
- 542 csv à traiter
- 280 Mo
- 370k commentaires
- 93k reply
- 433k lignes

Nettoyage des données

Principales étapes

- Utilisation de polars et pandas

- Nettoyage
  - Colonnes
  - Dates
  - Lignes dupliquées

- Ajout manuel
  - Genre
  - Vues
  - Subscribers
  - Rating

- Un seul csv avec toutes les données

Sentiment analysis

Principales étapes

- Utilisation de pandas, pytorch, transformers, nltk

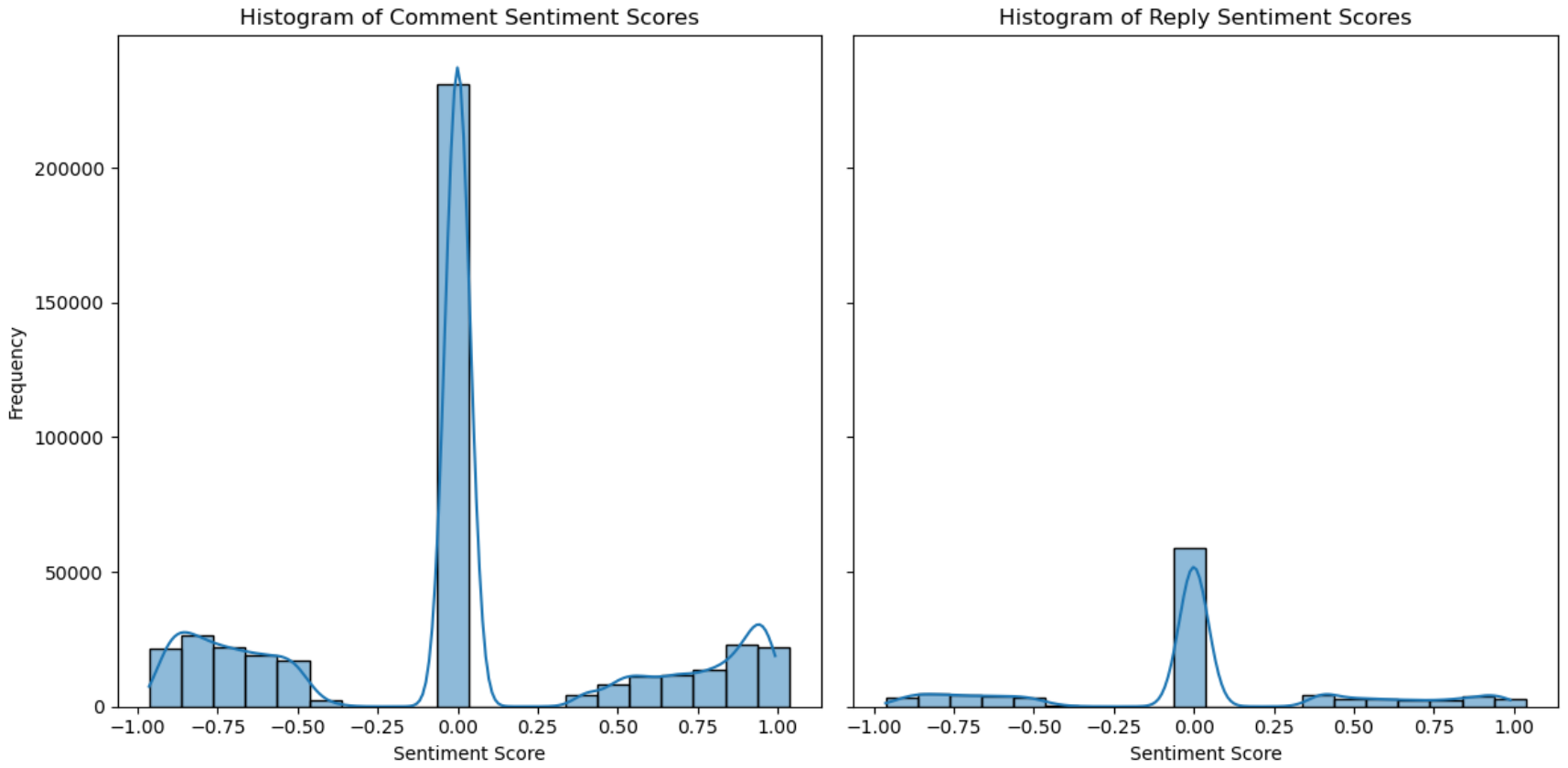
- Préparation du texte
  - Tokenize
  - Supprimer mots trop longs (30 caractères)
  - Contractions
  - Supprimer stop-words
  - Lemmatize

- Sentiment analysis
  - Modèle «cardiffnlp/twitter-roberta-base-sentiment-latest» état de l'art en sentiment analysis sur tweets
  - Utilisation de GPU NVIDIA
  - Echelle (-1,1) → négatif/neutre/positif

Sentiment analysis

Résultats

- La plupart des commentaires sont neutres
- Le modèle n’a pas le contexte du commentaire
- Anglais mal écrit
- Sarcasme/ironie mal analysés



Prédiction

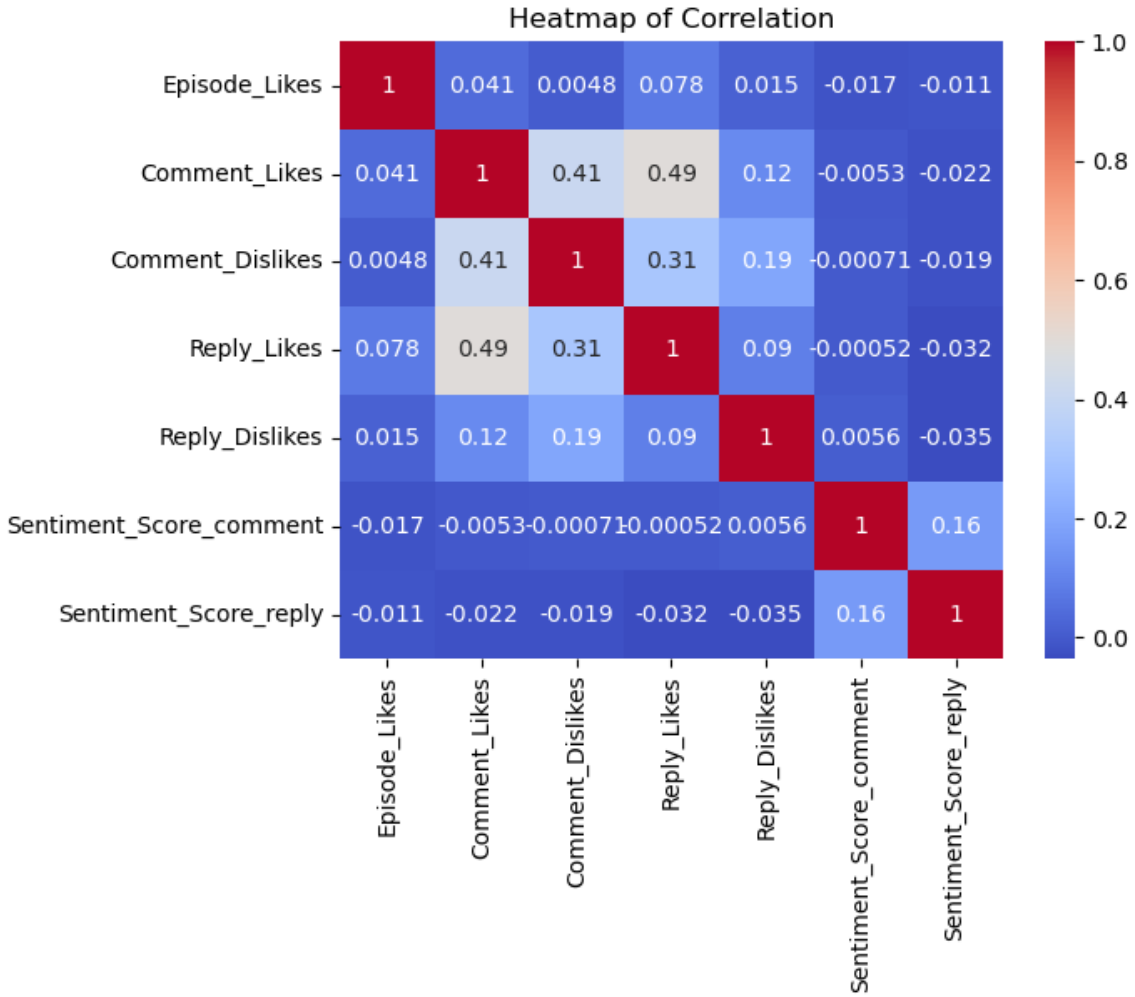
- Premier Model proposé
- Peu de donnée pour valider

Prédiction

Première approche

- Corrélation entre Comment_Likes, Comment_Dislikes, Reply_Likes

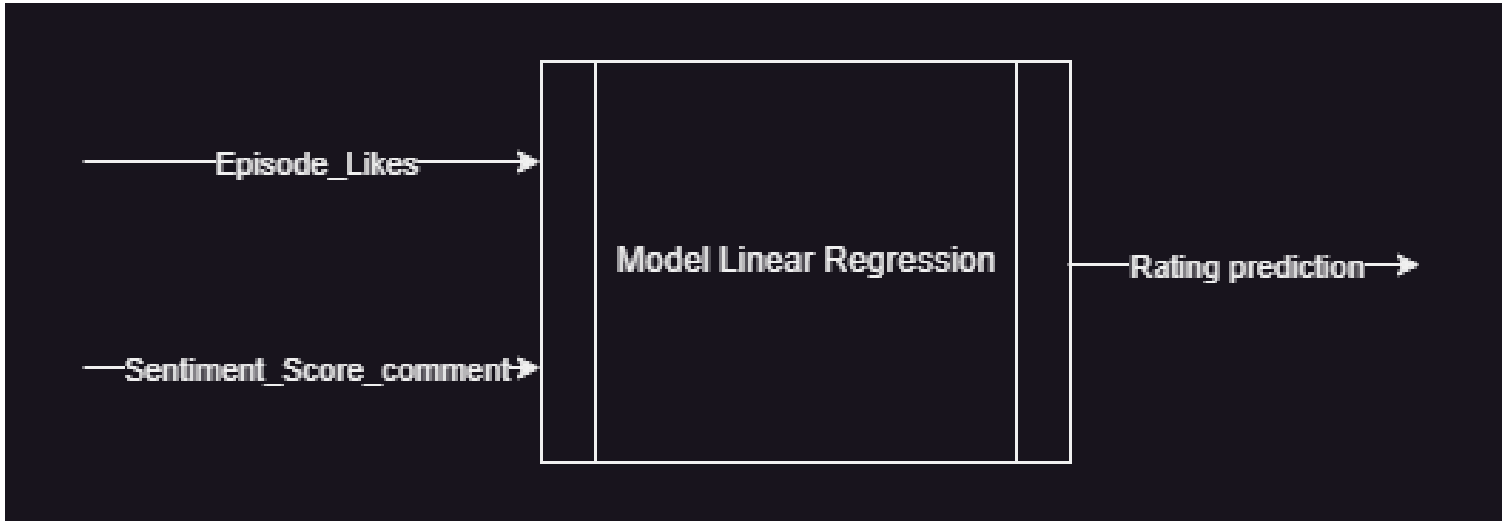
- Episode_Likes peu corrélée avec les autres inputs

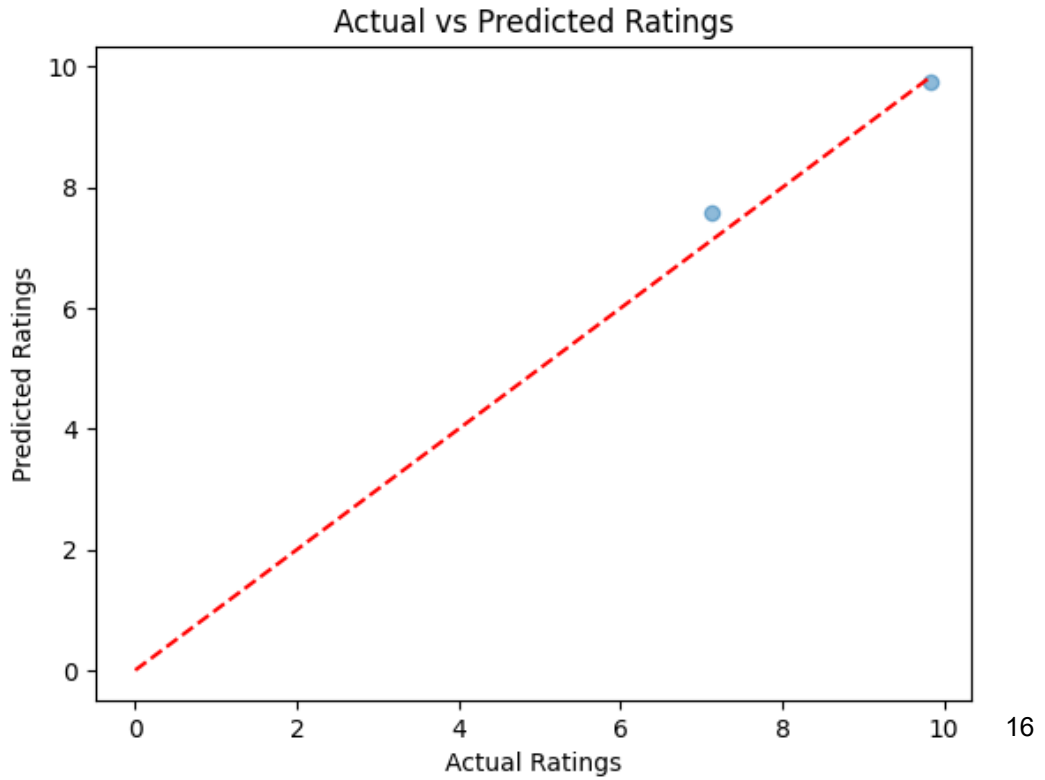


Prédiction

Model linear

- Bon résultat
- Trop peu de données
- Faible fiabilité

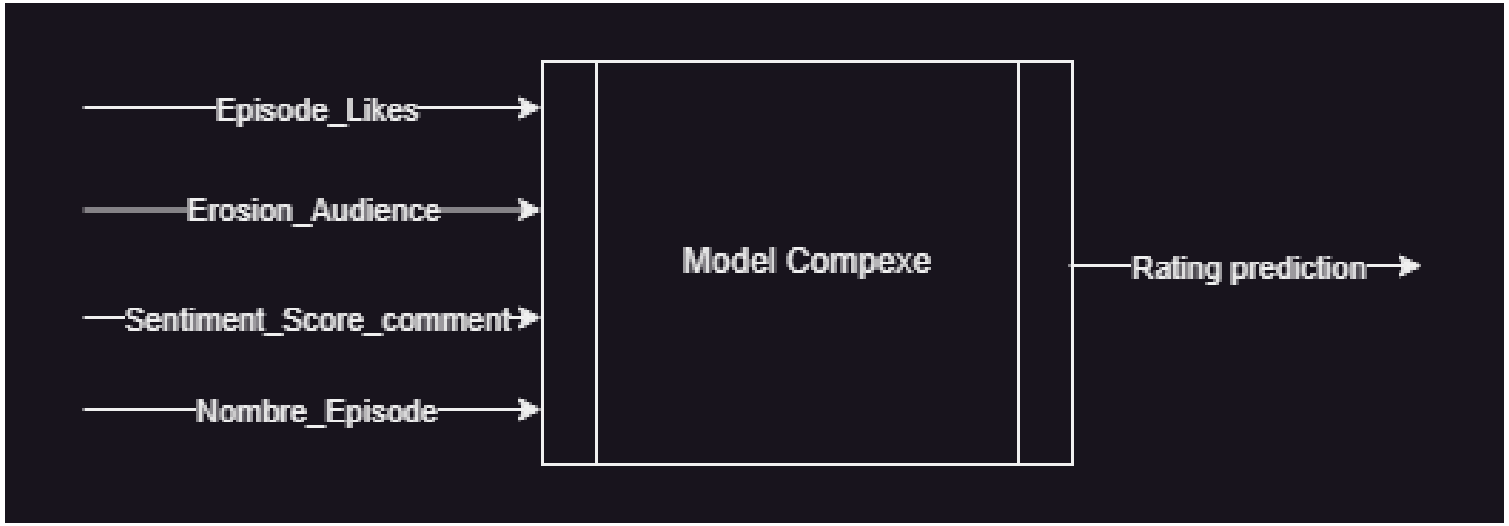


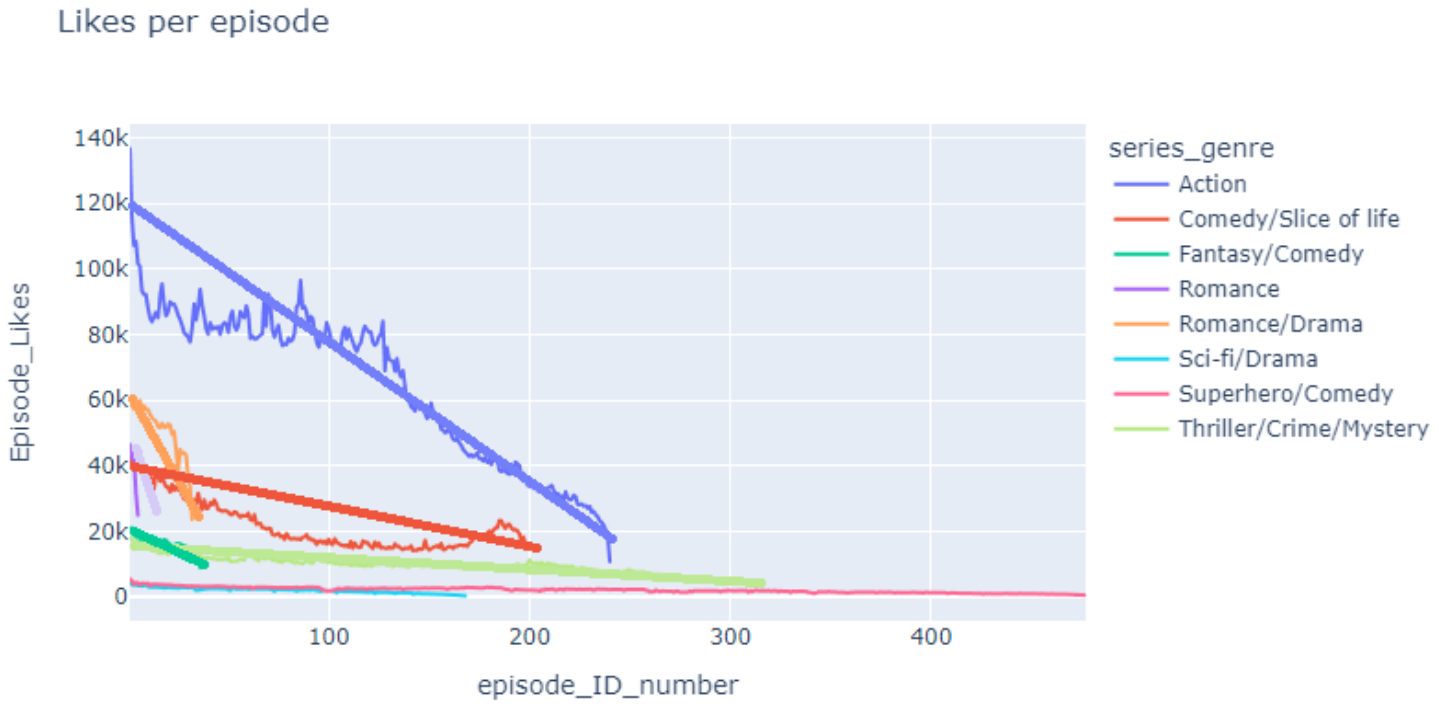


Prédiction

Model complex

- Ajout de l'érosion de l'audience comme indication
- Validation des commentaires positif/négatif avec le nombre de like/dislike
- Possibilité d'extraire également un gradient de sentiment pour avoir l'évolution des utilisateurs
- Avec plus de donnée le genre vas devenir important





Visualisation

Technologies utilisées

- Notebook pour tester les différents graphs

- Résultat final dans un script

Visualisation

Analyse des likes par épisode

- Isoler les épisodes

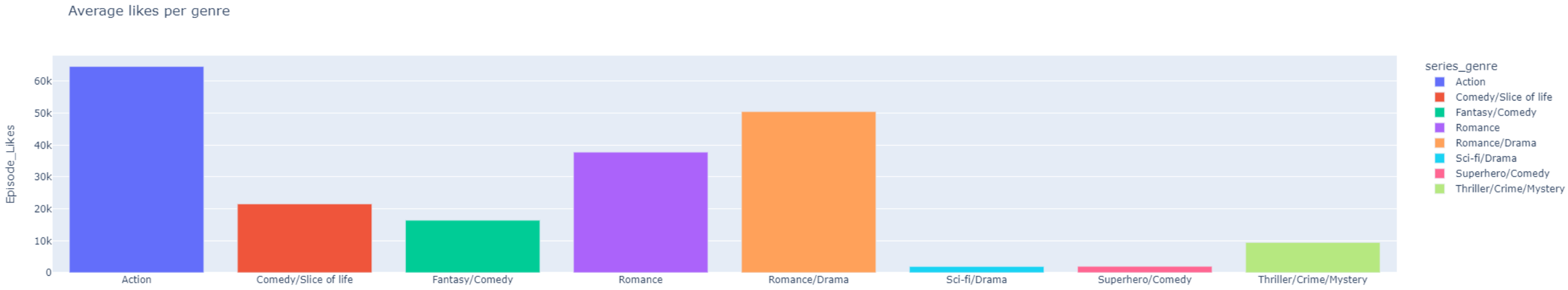
	series_genre	Episode_Date	Episode_Likes	episode_ID_number	Comment_Text
0	Action	2020-09-01	81653	100	I'm so glad Gray has grown and found peace for...
1	Action	2020-09-01	81653	100	I'm glad to see he's able to overcome his roof...
2	Action	2020-09-01	81653	100	Donald is playing 4D chess
3	Action	2020-09-01	81653	100	What is helmet up to these day Nothing much st...
4	Action	2020-09-01	81653	100	I really wish we can see Stephen again at some...
5	Action	2020-09-01	81653	100	Am I the only one who thinks the backpack is t...
6	Action	2020-09-01	81653	100	can we just take a moment and appreciate grays...
7	Action	2020-09-01	81653	100	I'm so happy that he's gotten over that fear a...
8	Action	2020-09-01	81653	100	My chest physically hurts again
9	Action	2020-09-01	81653	100	that mattress pin hiTS DEEP IN THE HEART BRO

	series_genre	Episode_Date	Episode_Likes	episode_ID_number
0	Action	2019-09-10	136613	1
1	Action	2019-09-10	115644	2
2	Action	2019-09-10	107151	3
3	Action	2019-09-17	108406	4
4	Action	2019-09-24	101632	5
5	Action	2019-10-01	100795	6
6	Action	2019-10-08	93428	7
7	Action	2019-10-15	92176	8
8	Action	2019-10-17	92229	9
9	Action	2019-10-22	88114	10

Visualisation

Analyse des likes moyens par genre

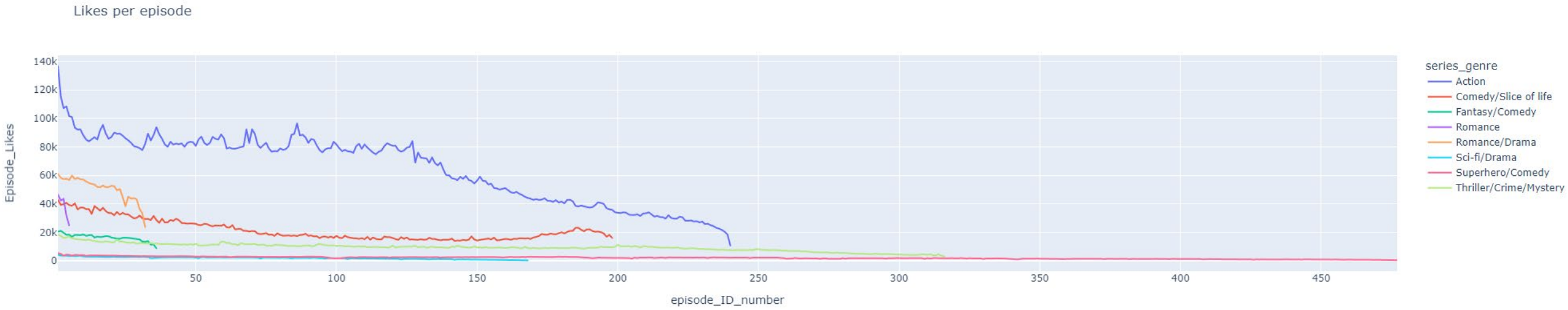
- Grouper par genre, puis faire la moyenne



Visualisation

Analyse des likes par épisode

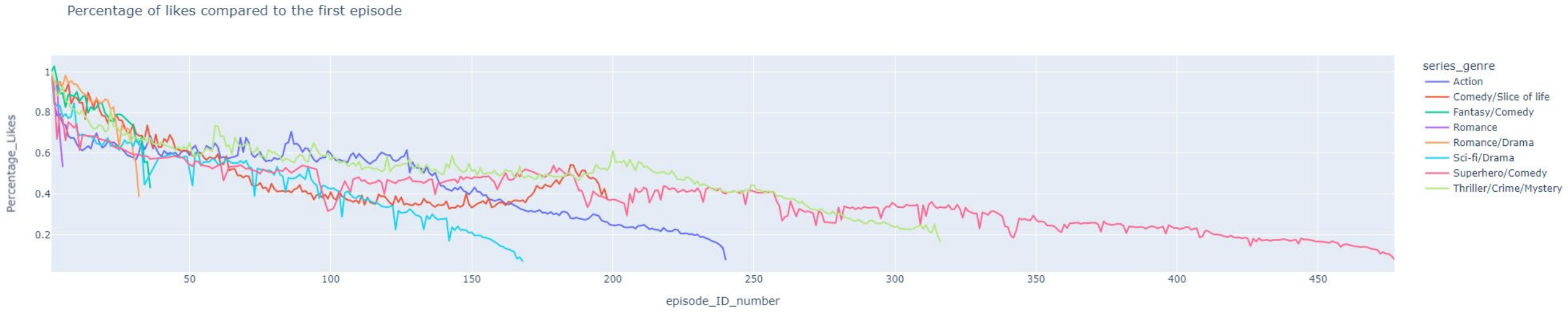
- Création du graphique à l'aide de plotly



Visualisation

Analyse du déclin des likes par genre

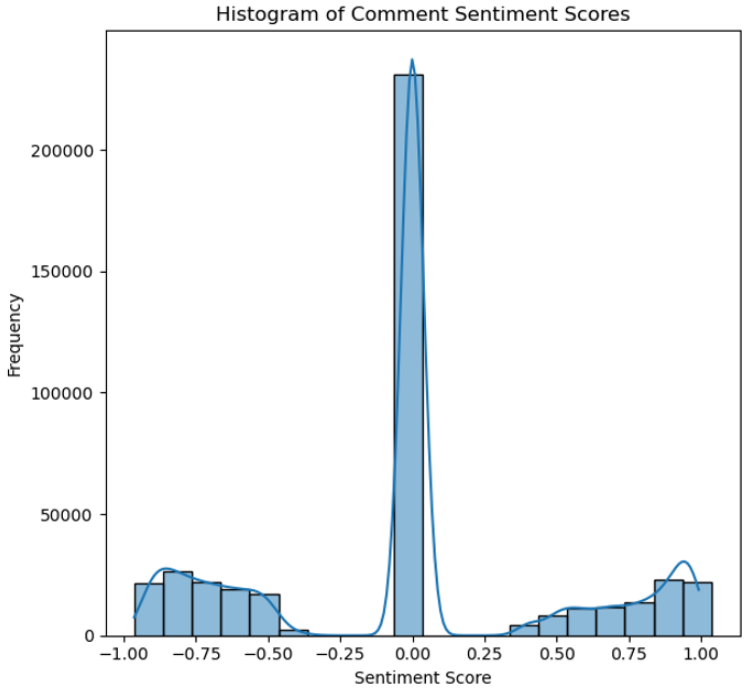
- Calculer le pourcentage de like d'un épisode par rapport au premier épisode de la série

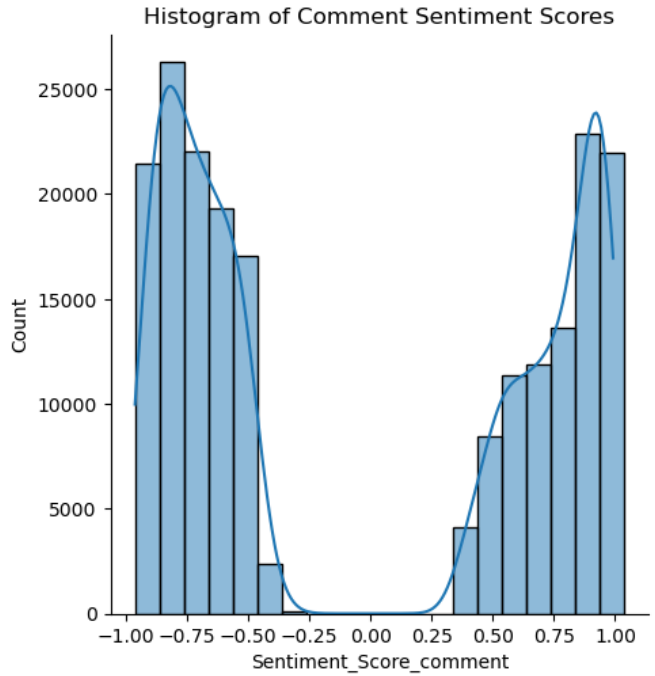


Visualisation

Analyse du sentiment par genre

- Drop les commentaires qui ont un sentiment score de 0

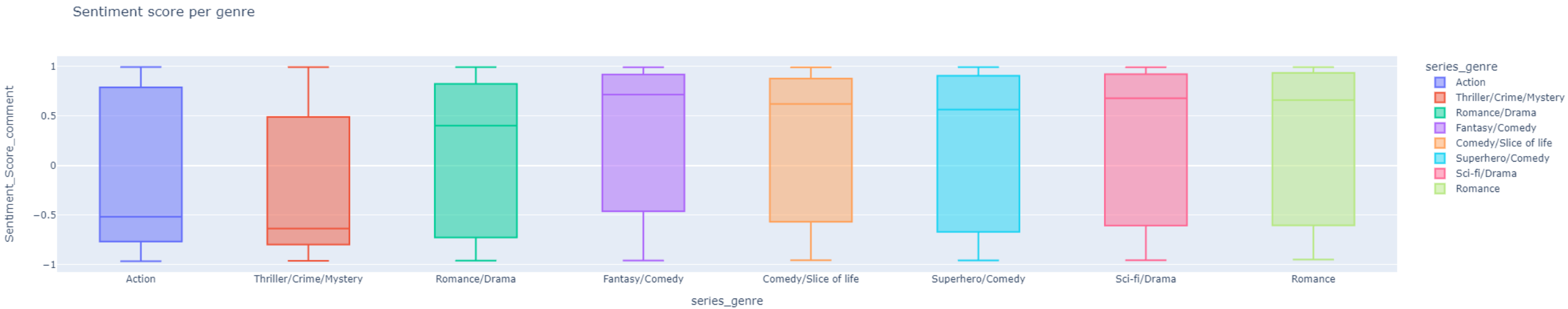




Visualisation

Analyse du sentiment par genre

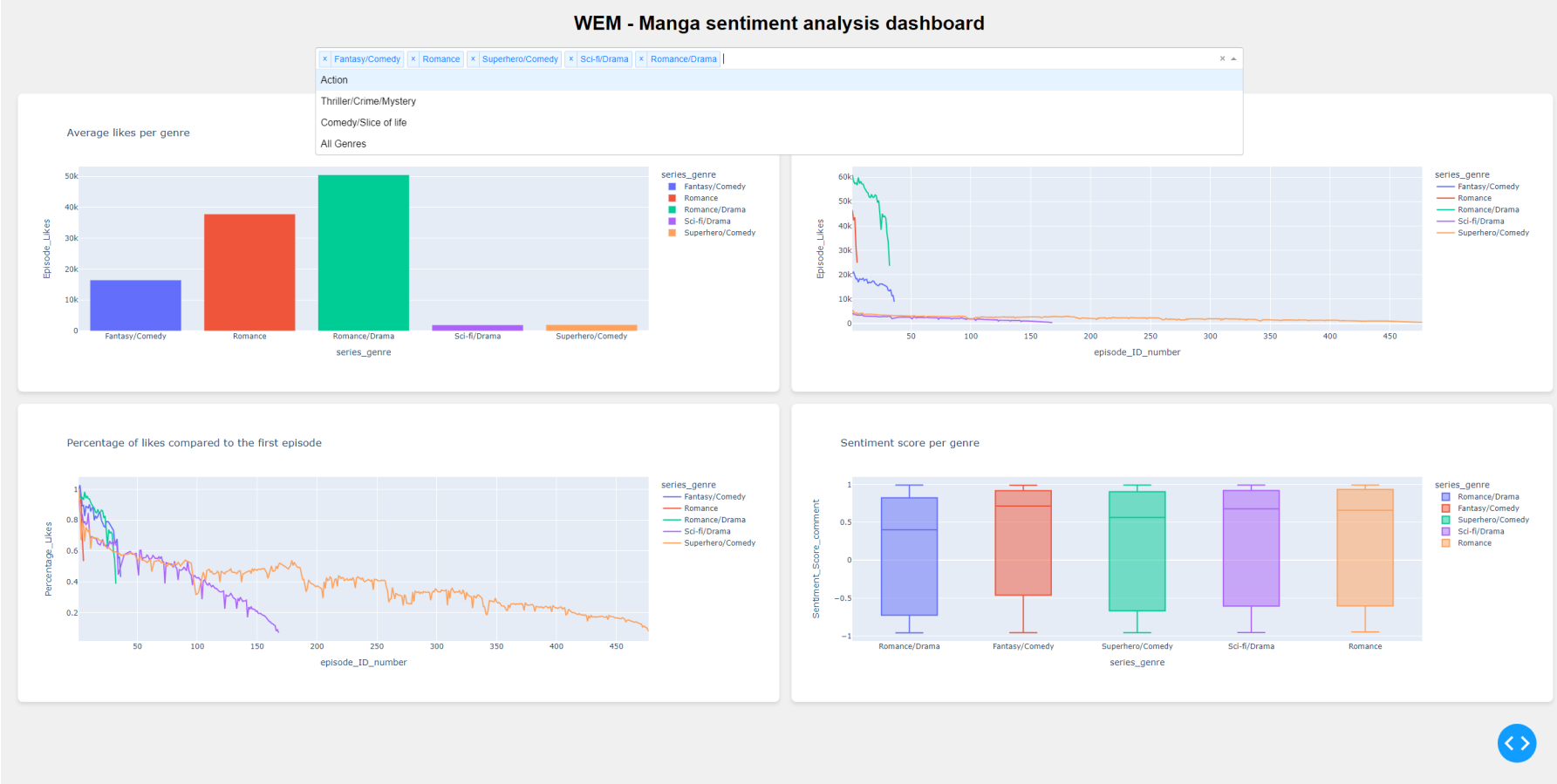
- Création de boxplots à l'aide de plotly



Visualisation

Création du dashboard

- Création de boxplots à l'aide de plotly



Conclusion

- Scraping
  - Très long
  - Code du site peut changer
- Nettoyage des données
  - Assez long d'ouvrir autant de csv avec pandas
- Sentiment analysis
  - Modèle «hugging face» facile à utiliser
  - Rapide avec un GPU
- Rating prédiction
- Visualisation
  - Très intéressant de tirer des graphs à partir d'un simple site web
  - Manque de données pour tirer de vraies conclusions

Pistes d'amélioration

- Général
  - Plus de séries, données obtenues insuffisantes
- Scraping
  - Code plus robuste et plus automatisé
  - Parallèlisation du scraping
- Nettoyage de données
  - Utiliser exclusivement polars
- Sentiment analysis
  - Mieux évaluer influence pré-traitement texte
- Prédication
  - Complexe features (time séries)
- Visualisation
  - Créer un index pour quantifier le déclin des likes par épisode

Merci!