Homework 2

AUTHOR
Thomas Zwiller

Task 1

We are going to return to the table of the top 100 wrestlers: https://www.cagematch.net/?id=2&view=statistics. Specifically, you are going to get the ratings/comments tables for each wrestler.

```
from bs4 import BeautifulSoup
import re
import numpy as np
import requests
import pandas as pd
from langdetect import detect, DetectorFactory

#lets start with our link
link = 'https://www.cagematch.net/?id=2&view=statistics'

#check to make sure the request is good
wrastling = requests.get(link)

#parse the content
wrastling_soup = BeautifulSoup(wrastling.content, 'html.parser')

#find all the rows I need
all_rows = wrastling_soup.find_all('tr')

#pull all the gimmicks on the page
wrastlers_list = wrastling_soup.select(
  '.LayoutContent .TableContents .TCol.TColSeparator a[href*="gimmick"]')

#flatten the links
link_list = [link['href'] for link in wrastlers_list]

#and then making them into a dataframe
wrestler_links = pd.DataFrame({'Links': link_list})

#now I'm going to extract all the wrestler codes from the links
wrestler_links['Code'] = wrestler_links['Links'].str.extract(r'(?<=nr=)(\d+)(?=&gimmick)'

#and then make the link start and end to concatenate them
link_start = 'https://www.cagematch.net/?id=2&nr='
link_end = '&page=99'

#setting up a temporary list to house the concatenated links
link_list = []
```

```
#and then creating all the links I'll need with a loop
for code in wrestler_links['Code']:
    link_code = code
    wrestler_link = link_start + link_code + link_end
    link_list.append(wrestler_link)

#making an empty dataframe
comment_df = pd.DataFrame()

#and then making a for loop which will iterate the link list
for link in link_list:
  #making a temporary frame that resets each loop
  rating_frame = []
  #making the request for the ratings
  ratins = requests.get(link)
  #parsing each link
  ratings_soup = BeautifulSoup(ratins.content, 'html.parser')
  #checking how many comments there are for each page
  ratings_rows = ratings_soup.find_all('div', class_='Comment')
  #then, for each comment, on the page, there's a sub-loop
  for rating in range(len(ratings_rows)):
    rating_row = {
      #that grabs the rating here
      'User': ratings_soup.select('.CommentHeader')[rating].text,
      #then the comment
      'Comment': ratings_soup.select('.CommentContents')[rating].text}
      #which are appened to the rating frame
    rating_frame.append(rating_row)
    #made into a pandas dict
    comment_frame = pd.DataFrame(rating_frame)
  #and then they are concatenated to the comment df
  comment_df = pd.concat([comment_df, comment_frame], ignore_index=True)
```

With the data now read in, I want to clean it using regex.

```
#using regex to pull the rating out of the comment
comment_df['Rating'] = comment_df['Comment'].str.extract(r'(\d+\.\d|\d\.\d)').astype(floa

#using regex to pull the date from the comment
comment_df['Date'] = comment_df['User'].str.extract(r'(\d\d\.\d\d\.\d\d\d\d)')

#using regex to get rid of the date
comment_df['User'] = comment_df['User'].str.replace(r' wrote on \d\d\.\d\d\.\d\d\d\d\:' ,

#using regex to get rid of the date
comment_df['Comment'] = comment_df['Comment'].str.replace(r'(\[\d+\.\d+\]|\[\d\.\d\])', '

#this turns any links into 'DELETE'
comment_df['Comment'] = comment_df['Comment'].str.replace(r'http\S*', 'DELETE', regex=Tru

#and then I drop the previosuly existing links
```

```
comment_df = comment_df[comment_df['Comment'] != 'DELETE'].reset_index(drop=True)

#and then I'm checking for the comment of each language
comment_df['Comment Lang'] = comment_df['Comment'].apply(lambda x: detect(x) if isinstanc

#if it's English, it stays in, else it goes
comment_df = comment_df[comment_df['Comment Lang'] == 'en']

#and if it's na, it's getting dropped
comment_df = comment_df.dropna(subset=['Comment'])

comment_df.head(10)
```

	User	Comment	Rating	Date	Comment Lang
0	JediSaiyanMaster1203	"As I'm writing this comment, Kenta Kobashi i...	10.0	29.04.2025	en
1	Conor	"As far as I'm concerned, Kobashi is the grea...	10.0	04.04.2025	en
2	gabsdavero	"A pure wrestler. Technical mastermind. Had t...	10.0	29.03.2025	en
3	NintendoTS	"Greatest babyface of all time. Absolutely ins...	NaN	27.03.2025	en
4	Brutish Dandy	"Obligatory ten-outta-ten for Kenta Kobashi, ...	10.0	27.03.2025	en
5	Lalo Campos	"The greatest wrestler of all time, his match...	10.0	26.02.2025	en
6	Porzingi	"Kenta Kobashi's chops are my favorite of all...	10.0	15.02.2025	en
8	anarchovamp	"John Bradshaw famously dubbed himself a wres...	10.0	27.12.2024	en
9	ZephsPancakes	"I personally slightly edge Misawa over Kobas...	10.0	12.12.2024	en
10	TPG	"If I'm being honest, Kobashi's probably one ...	10.0	05.12.2024	en

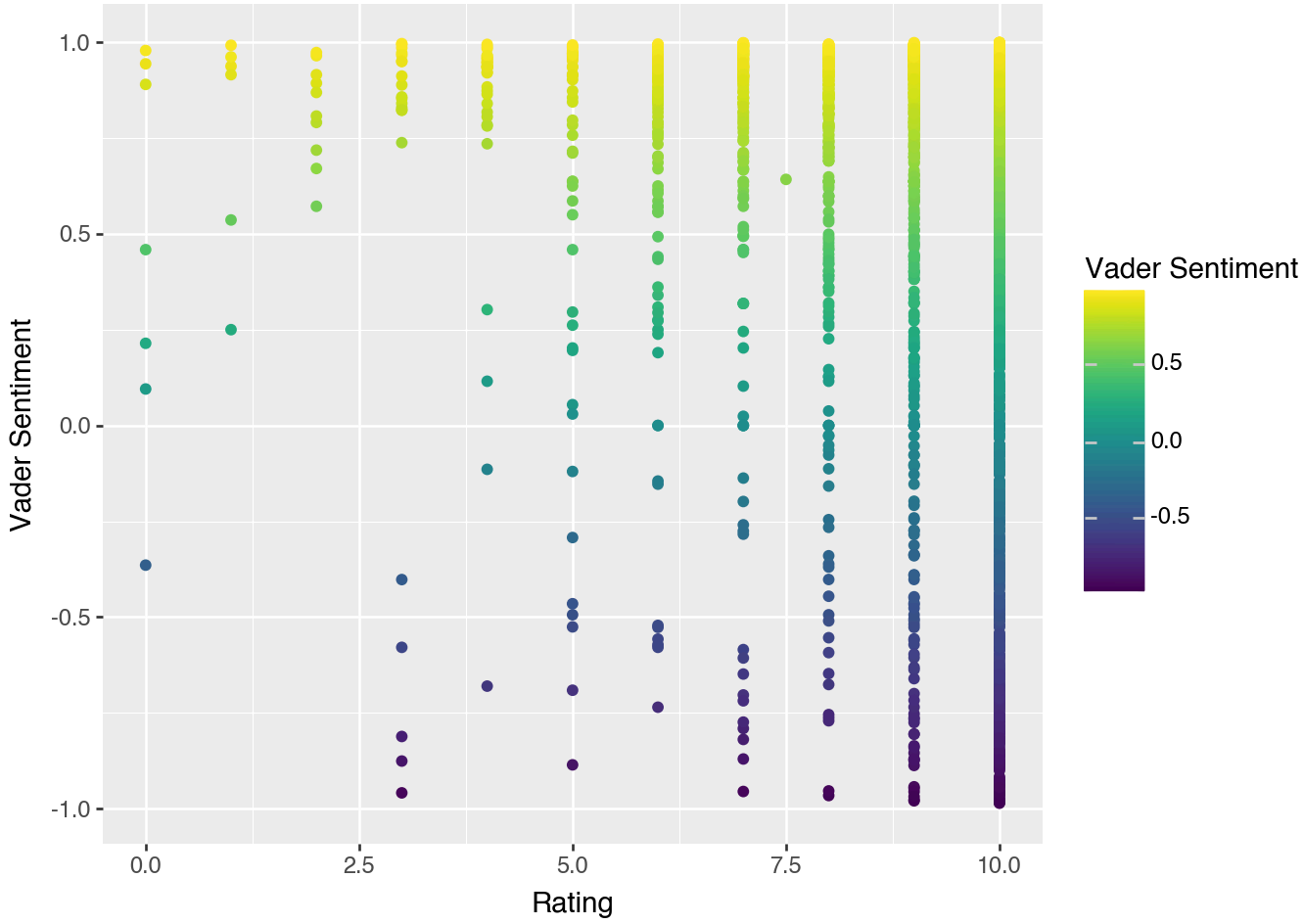
Task 2

Perform any form of sentiment analysis. What is the relationship between a reviewer's sentiment and their rating?

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from plotnine import ggplot, geom_point, aes, stat_smooth, facet_wrap
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

#starting by initializing the sentiment analyzer
vader = SentimentIntensityAnalyzer()

#looping through the comments in order to assign a sentiment score to a new column
for comment in range(len(comment_df)):
  sentiment_score = vader.polarity_scores(comment_df.iloc[comment]['Comment']).get('compo
```

```
  comment_df.loc[comment, 'Vader Sentiment'] = sentiment_score

#getting the correlation
ratings_corr = comment_df['Rating'].corr(comment_df['Vader Sentiment'])

#and then plotting
vader_plot = ggplot(comment_df, aes("Rating", "Vader Sentiment", color= "Vader Sentiment"

vader_plot
```

```
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-
packages/plotnine/layer.py:364: PlotnineWarning: geom_point : Removed 8571 rows
containing missing values.
```



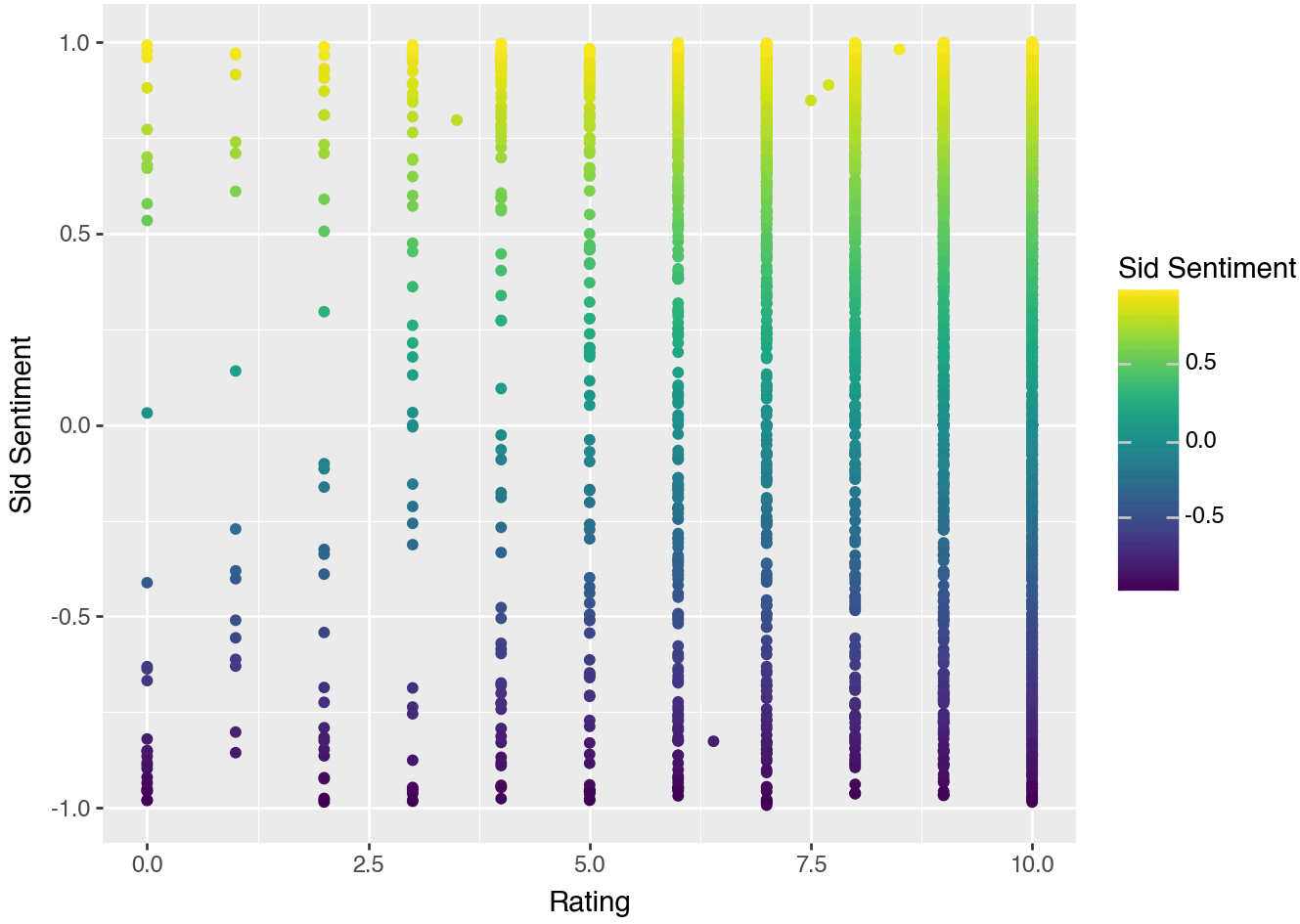
Vader Correlation: `r ratings_corr`

I didn't love the overall result from the Vader sentiment. The distribution on the comments with a rating of 10 was much too wide; it's hard to see a lot of 10 ratings being incredibly negative. So I wanted to try something else

```
#initializng the sentiment intensity analyzer
sid = SentimentIntensityAnalyzer()

#dropping the na values
```

```
comment_df = comment_df.dropna(subset=['Comment'])

#appling sid to the comments and writing it out to a new column
comment_df['Sid Sentiment'] = comment_df['Comment'].apply(lambda x: sid.polarity_scores(x

#getting the correlation
ratings_corr_2 = comment_df['Rating'].corr(comment_df['Sid Sentiment'])

#plotting
sid_plot = ggplot(comment_df, aes("Rating", "Sid Sentiment", color= "Sid Sentiment")) + g

sid_plot
```

```
/var/folders/dc/30zw0xgj67z2hzky0sd_pkv80000gp/T/ipykernel_61536/2079831934.py:8:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-
packages/plotnine/layer.py:364: PlotnineWarning: geom_point : Removed 386 rows containing
missing values.
```

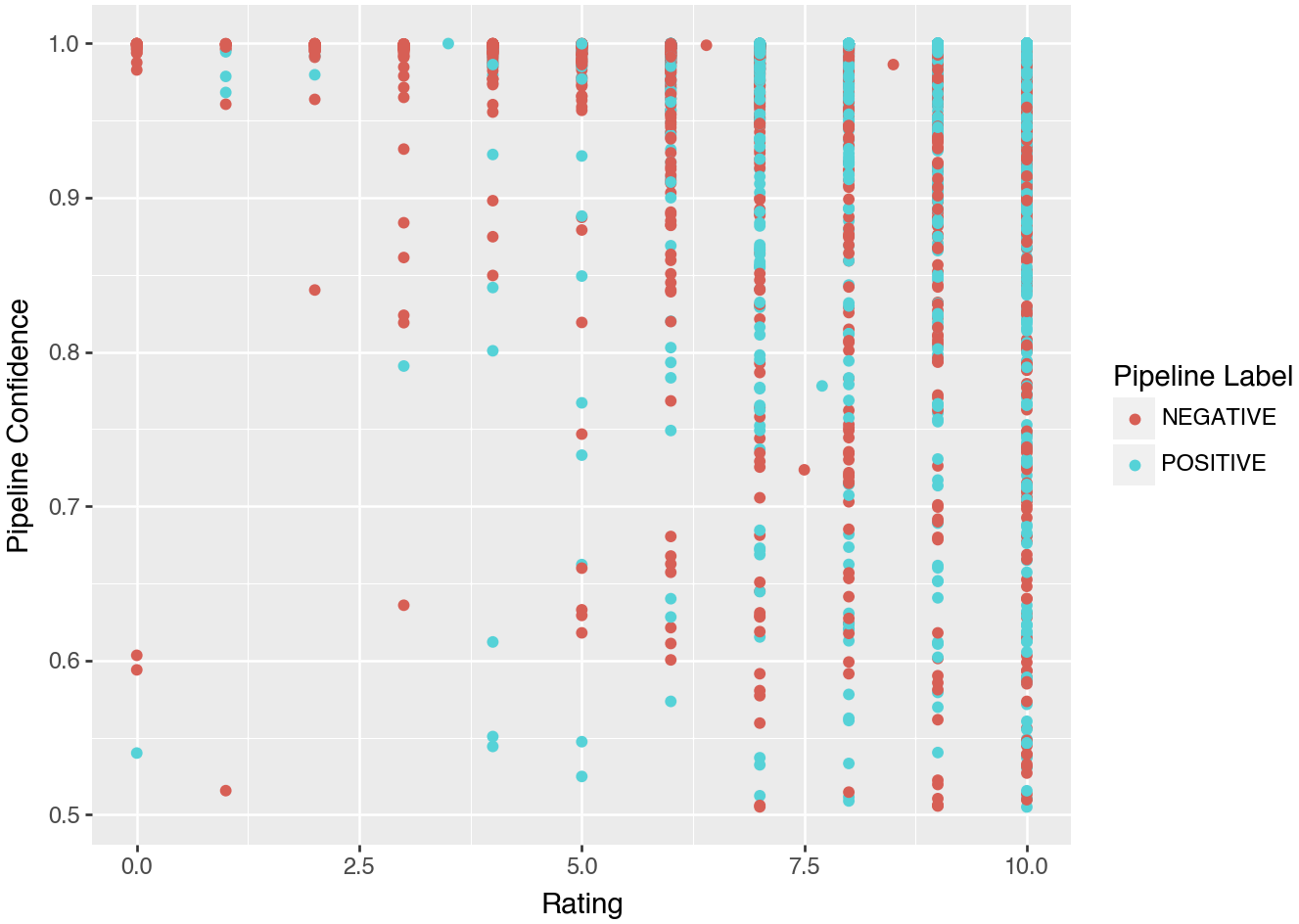


Sid Correlation: `r ratings_corr_2`

Unfortunately, the sid analyzer didn't really seem to do much better so I decided to also use a transformer

```
from transformers import pipeline

#maxing the length at 512
sentiment_analysis = pipeline("sentiment-analysis", truncation=True, max_length=512)

#applying the analyzer and reading out the score
comment_df['Pipeline Confidence'] = comment_df['Comment'].apply(lambda x: sentiment_analy

#applying the analyzer and reading out the label
comment_df['Pipeline Label'] = comment_df['Comment'].apply(lambda x: sentiment_analysis(x

#checking the correlation
ratings_corr_3 = comment_df['Rating'].corr(comment_df['Pipeline Confidence'])

#and then plotting
pipe_plot = ggplot(comment_df, aes("Rating", "Pipeline Confidence", color= "Pipeline Labe

pipe_plot
```

```
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-
packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update jupyter and
ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html
No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-
english and revision 714eb0f (https://huggingface.co/distilbert/distilbert-base-uncased-
finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is not
recommended.
Device set to use mps:0
/var/folders/dc/30zw0xgj67z2hzky0sd_pkv80000gp/T/ipykernel_61536/3482830205.py:7:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/var/folders/dc/30zw0xgj67z2hzky0sd_pkv80000gp/T/ipykernel_61536/3482830205.py:10:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-
packages/plotnine/layer.py:364: PlotnineWarning: geom_point : Removed 386 rows containing
missing values.
```

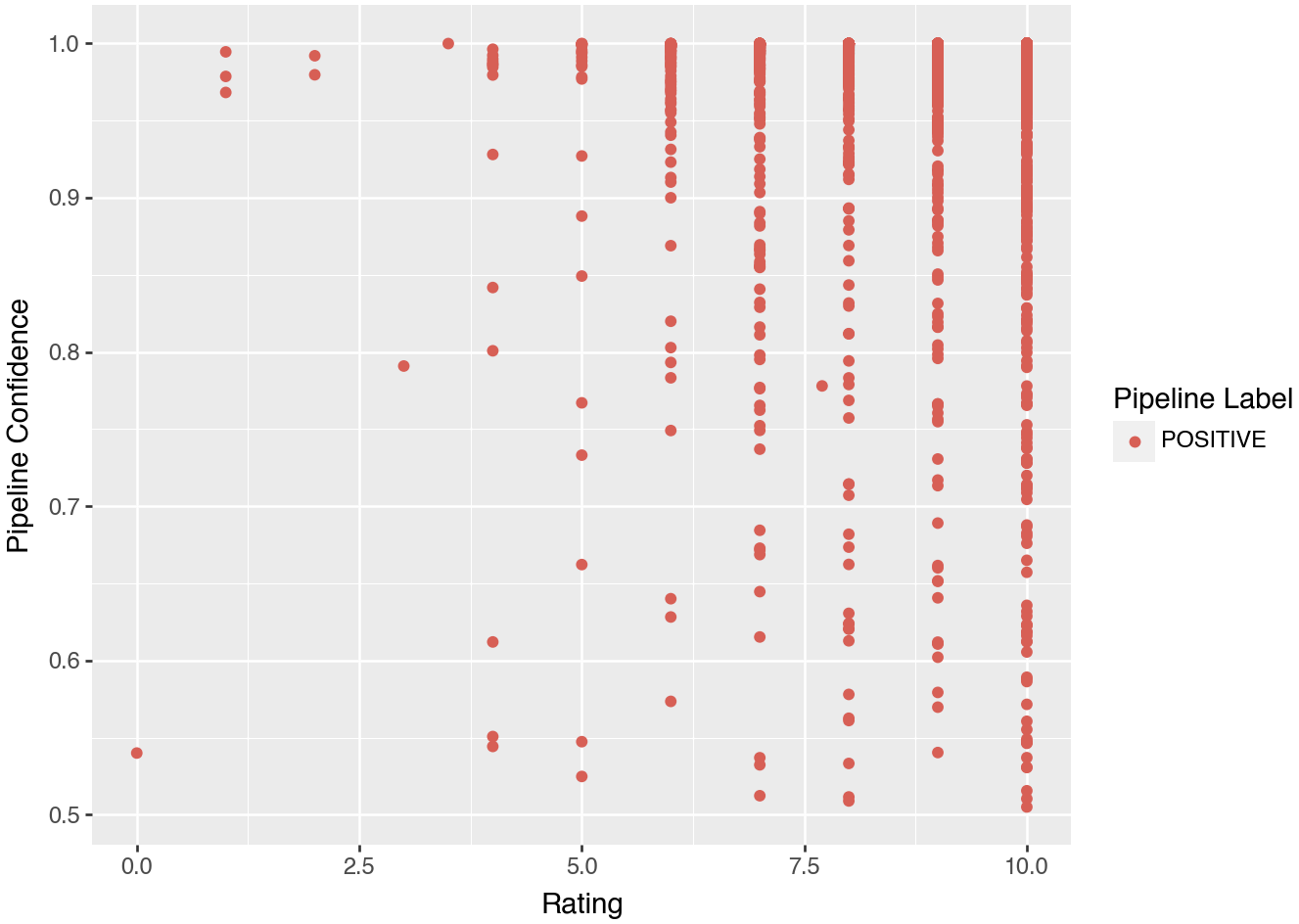


Pipeline Correlation: `r ratings_corr_3`

The chart, while interesting, needed to get cleaned up a little bit. Too busy.

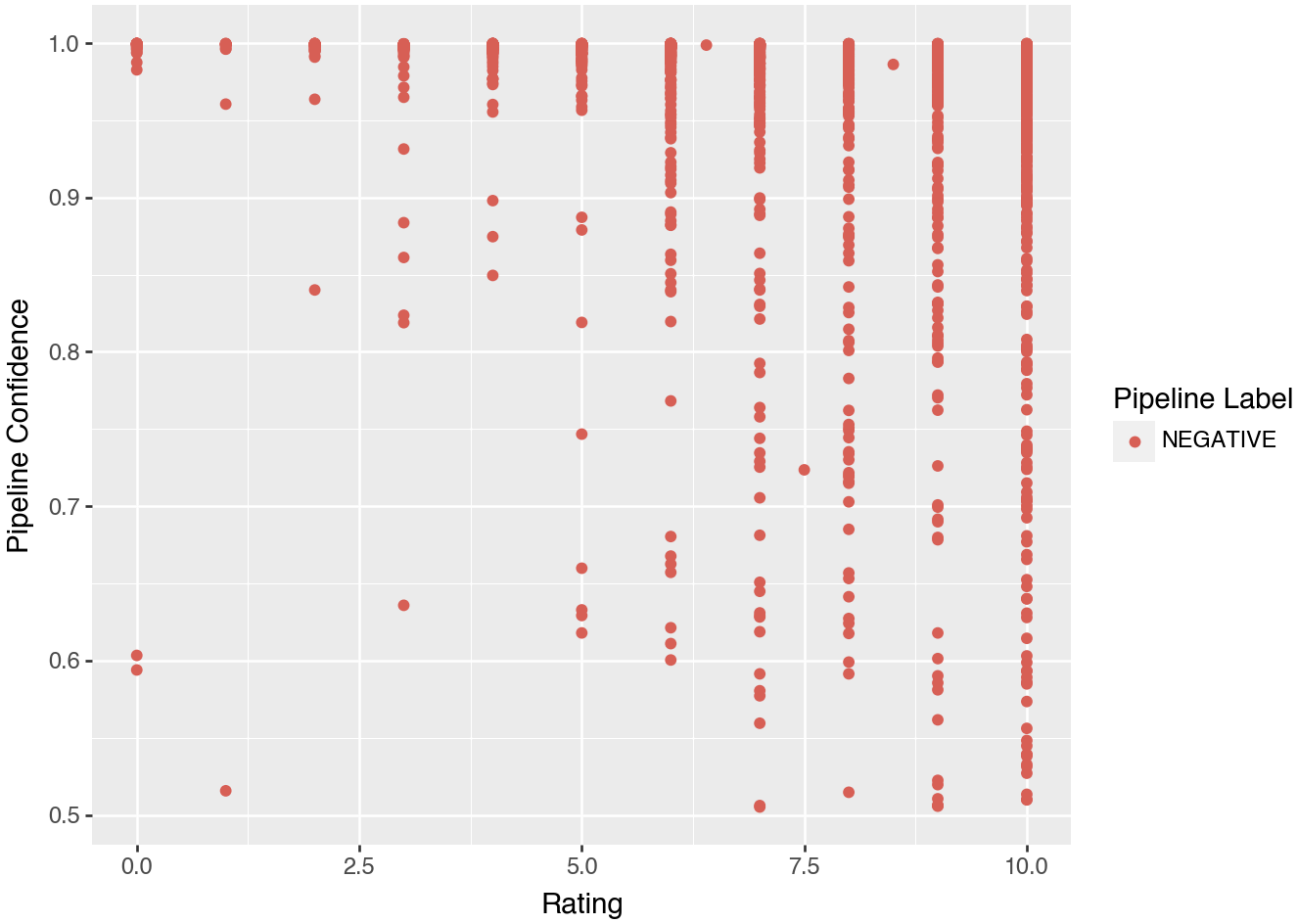
```
#poistive only
pos_df = comment_df[comment_df['Pipeline Label'] == 'POSITIVE']

#plot
pos_plot = ggplot(pos_df, aes("Rating", "Pipeline Confidence", color= "Pipeline Label"))

pos_plot
```

```
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-
packages/plotnine/layer.py:364: PlotnineWarning: geom_point : Removed 330 rows containing
missing values.
```




```
#negative only
neg_df = comment_df[comment_df['Pipeline Label'] == 'NEGATIVE']

#plot
neg_plot = ggplot(neg_df, aes("Rating", "Pipeline Confidence", color= "Pipeline Label"))

neg_plot
```

```
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-
packages/plotnine/layer.py:364: PlotnineWarning: geom_point : Removed 56 rows containing
missing values.
```



I think that splitting out the charts reveals an interesting pattern. The lower ratings had more negative comments, and the model seemed to be much more confident in assigning that rating.

However, the same problem as before seemed to pop up once the model got to a rating of 6 and beyond. I suspect that it's largely because comments do contain legitimate critiques or might be talking about fights in which the fighter lost, which could be seen as negative.

But I decided to see the actual count of each and found that despite the fact that there seemed to be a lot of negative comments, there were actually quite a few more positive comments, it was just hard to render them all on the plot.

```
#groupping the pos ratings so I can count how many there are
pos_nums = pos_df.groupby(['Rating','Pipeline Label']).size()

corr_pos = pos_df['Rating'].corr(pos_df['Pipeline Confidence'])

print(pos_nums)

#groupping the neg ratings so I can count how many there are
neg_nums = neg_df.groupby(['Rating','Pipeline Label']).size()

corr_neg = neg_df['Rating'].corr(neg_df['Pipeline Confidence'])
```

```
print(neg_nums)
```

```
Rating  Pipeline Label
0.0     POSITIVE             1
1.0     POSITIVE             3
2.0     POSITIVE             2
3.0     POSITIVE             1
3.5     POSITIVE             1
4.0     POSITIVE            14
5.0     POSITIVE            23
6.0     POSITIVE            84
7.0     POSITIVE           230
7.7     POSITIVE             1
8.0     POSITIVE           475
9.0     POSITIVE          1182
10.0    POSITIVE          5315
dtype: int64
Rating  Pipeline Label
0.0     NEGATIVE            28
1.0     NEGATIVE            13
2.0     NEGATIVE            31
3.0     NEGATIVE            45
4.0     NEGATIVE            70
5.0     NEGATIVE            95
6.0     NEGATIVE           188
6.4     NEGATIVE             1
7.0     NEGATIVE           210
7.5     NEGATIVE             1
8.0     NEGATIVE           213
8.5     NEGATIVE             1
9.0     NEGATIVE           219
10.0    NEGATIVE           356
dtype: int64
```

Positive Correlation: `r corr_pos` Negative Correlation: `r corr_neg`

So I would say, in general:

When a comment had a lower rating, the model had much less confidence it in the assessment that it was a positive comment, but seemed to be more confident with it being a negative comment.

The model did struggle to differeniate betwen the higher ratings, but I do think it's because the language included could be nuanced and contain valid critiques but still result in a positive overall sentiment that the computer was unable to detect.

By and large, the positive comments were much more numerous than the negative comments, which was hard to extrapolate from the plot.

Task 3

Perform any type of topic modeling on the comments. What are the main topics of the comments? How can you use those topics to understand what people value?

```
from bertopic import BERTopic
from bertopic.vectorizers import ClassTfidfTransformer

#reducing the frequently found words for the topic analyzer
ctfidf_model = ClassTfidfTransformer(
  reduce_frequent_words=True
)

#setting up the topic model
topic_model = BERTopic(ctfidf_model=ctfidf_model)

#getting the topics and probabilities
topics, probs = topic_model.fit_transform(comment_df['Comment'].to_list())

#making it an object
model_df = pd.DataFrame(topic_model.get_topic_info())

model_df.head(10)
```

```
huggingface/tokenizers: The current process just got forked, after parallelism has
already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
    - Avoid using `tokenizers` before the fork if possible
    - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
OMP: Info #276: omp_set_nested routine deprecated, please use omp_set_max_active_levels
instead.
```

	Topic	Count	Name	Representation	Representative_Docs
0	-1	1929	-1_mic_bryan_skills_pro	[mic, bryan, skills, pro, danielson, technical...	["There has been a debate within the wrestlin...
1	0	261	0_punk_cm_punks_criteria	[punk, cm, punks, criteria, 2011, pipebomb, su...	["A mixed bag is the best way to put Punk's c...
2	1	238	1_jericho_chris_jerichos_y2j	[jericho, chris, jerichos, y2j, wcw, reinvent,...	["Chris Jericho is an all time great wrestler...
3	2	225	2_undertaker_taker_deadman_streak	[undertaker, taker, deadman, streak, gimmick, ...	["One of greatest of all time. He is most res...
4	3	202	3_aj_styles_tna_phenomenal	[aj, styles, tna, phenomenal, club, ajs, bulle...	["AJ Styles will always be one of my favourit...

	Topic	Count	Name	Representation	Representative_Docs
5	4	184	4_ospreay_ospreays_ricochet_flippy	[ospreay, ospreays, ricochet, flippy, will, mo...	["In a great consideration to his talents ins...
6	5	153	5_female_shes_womens_she	[female, shes, womens, she, her, women, woman,...	["The best female wrestler of all time, and o...
7	6	133	6_okada_okadas_rainmaker_tanahashi	[okada, okadas, rainmaker, tanahashi, iwgp, ex...	["Okada from 2012 up until the end of 2013 I ...
8	7	131	7_austin_cold_stone_steve	[austin, cold, stone, steve, 316, attitude, be...	["Stone Cold Steve Austin pushed the boundres...
9	8	123	8_shawn_michaels_rockers_heartbreak	[shawn, michaels, rockers, heartbreak, dx, sha...	["Shawn Michaels is debatably the greatest wr...

In general, the model seemed to group topics by the wrestler that the comment was about, which makes sense, as the comments would have included their names and then similar words to describe how they felt, as well as a simialr sentiment, as the general consensus was positivity.

I think the best way to look at these models would be to see which wrestler (or wrestlers) it’s about and then how that wrestler is described to better understand their brand.