IDL TP Final `scikit-learn` Projet

Charger les donnés

In [1]:
```
import os
import pandas as pd
```

In [2]:
```
def load_reviews(data_dir):
        reviews = []
        labels = []
        for label in ["pos", "neg"]:
                directory = os.path.join(data_dir, label)
                for filename in os.listdir(directory):
                        if filename.endswith(".txt"):
                                file_path = os.path.join(directory, filename
                                with open(file_path, 'r', encoding='utf-8')
                                        reviews.append(file.read())
                                        labels.append(1 if label == "pos" el
        return reviews, labels
```

In [3]:
```
data_dir = 'imdb_smol'
reviews, labels = load_reviews(data_dir)
reviews_df = pd.DataFrame({'review': reviews, 'label': labels})
```

In [4]:
```
print("Nombre de données:", len(reviews_df))
```

```
Nombre de données: 602
```

In [5]:
```
reviews_df.head()
```

Out[5]:

	review	label
0	The production quality, cast, premise, authent...	1
1	This is no art-house film, it's mainstream ent...	1
2	Two great comedians in a great Neil Simon movi...	1
3	I'm a fan of TV movies in general and this was...	1
4	Once upon a time in a castle...... Two little ...	1

In [6]:
```
print("\nNombre de notes positives:", (reviews_df['label'] == 1).sum())
print("Nombre de notes négatives:", (reviews_df['label'] == 0).sum())
```

```
Nombre de notes positives: 301
Nombre de notes négatives: 301
```

In [7]:
```
print("\nInformations générales sur le DataFrame:")
print(reviews_df.info())
```

```
Informations générales sur le DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 602 entries, 0 to 601
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   review  602 non-null    object
 1   label   602 non-null    int64 
dtypes: int64(1), object(1)
memory usage: 9.5+ KB
None
```

In [8]:

```
for index, row in reviews_df.sample(5).iterrows():
        print("\nIndex de la note de review:", index, "\nTexte de la note de
        print("Étiquette:", "Positive" if row['label'] == 1 else "Négative")

        print("\nIndex de la note de review:", index, "\nTexte de la note de
        print("Étiquette:", "Positive" if row['label'] == 1 else "Négative")
```

Index de la note de review: 307
Texte de la note de review:
 Possible Spoiler

'Return to Cabin by the Lake' is a useless mo
vie. The acting was not good and the plot wasn't even remotely interesting.<
br />
'Cabin by the Lake' is a good TV movie. The sequel was not. Judd
Nelson was very good in the first film and put a whole lot more into his cha
racter than in this. It seemed as if HE wasn't even interested in doing the
sequel. His acting was good but it could have been better. I really don't wa
nt to comment on the rest of the cast because in my opinion, they're not eve
n worth mentioning. But I'll do it. The character of Alison isn't even hardl
y shown in the first part of the film. All of a sudden she's the center of a
ttention next to Stanley Caldwell. The role didn't make sense and it should
have been thought out a little better. Dahlia Salem was absolutely terrible.
Her acting was way below decent and the casting people should have looked fo
r somebody else, anybody else. The director, Mike, was a confusing characte
r. He seemed to have a purpose for being there but it didn't seem like his d
eath was necessary. The acting for this role was good, nothing great but bet
ter than Salem's.

The plot was real lousy if you think about it.
Stanley, who is presumed dead, makes his way onto the set of 'Cabin by the L
ake', the movie based on his script. He stumbles upon the director and in a
short time, the director is dead and Stanley is running the show. Yeah, out
of nowhere the crew is just going to let this stranger come into the picture
and finish the film not knowing anything about him. There's some killings, n
ot a whole lot, and the one's that are shown are ridiculous. One of the actr
esses on the set gets electrocuted while filming a scene. Another character
gets chewed up by a motorboat. And one gets tangled up in a plant before dro
wning. These writers must have been hard up for excitement.

I jus
t have to say that I was not impressed with the filming of the movie. The wa
y that it kept changing from looking low budget back to normal started to be
come irritating very fast. Also, the new cabin by the lake was poorly create
d. We aren't shown it but only in a few scenes, and the whole thing with the
chain in the basement was useless. It worked in the first film only because
we were shown the room a lot more, but it didn't work in this one.

There were too many characters in this sequel. All of them except for a fe
w had no reason to be there. The acting of what little is showed was really
bad and...they just didn't have a purpose in this movie.

All in a
ll, 'Return to Cabin by the Lake' is a sequel picking up from where the firs
t left off. 'Cabin by the Lake' I can take but this was just not impressive.
Judd Nelson should have avoided this one and so should you. It's nothing lik
e the first and it went entirely too slow. Nothing happened in the first hou
r and it continued to drag on for the second. Not to mention that the writin
g was horrible. Put this on only if you need some help getting to sleep.

So, we see that Stanley defies death and is still alive and well. By
the way he talks, it sounds like there could be a possible third installment
to a movie good just by itself. Quit throwing in sequels and we may be alrig
ht!

(Did the film makers not realize that they showed us how they
filmed the lake scenes from the first one? They were all done in a tank. Nev
er, never reveal the secrets of filming.)

Étiquette: Négative

Index de la note de review: 307
Texte de la note de review:
 Possible Spoiler

'Return to Cabin by the Lake' is a useless mo
vie. The acting was not good and the plot wasn't even remotely interesting.<
br />
'Cabin by the Lake' is a good TV movie. The sequel was not. Judd
Nelson was very good in the first film and put a whole lot more into his cha

racter than in this. It seemed as if HE wasn't even interested in doing the
sequel. His acting was good but it could have been better. I really don't wa
nt to comment on the rest of the cast because in my opinion, they're not eve
n worth mentioning. But I'll do it. The character of Alison isn't even hardl
y shown in the first part of the film. All of a sudden she's the center of a
ttention next to Stanley Caldwell. The role didn't make sense and it should
have been thought out a little better. Dahlia Salem was absolutely terrible.
Her acting was way below decent and the casting people should have looked fo
r somebody else, anybody else. The director, Mike, was a confusing characte
r. He seemed to have a purpose for being there but it didn't seem like his d
eath was necessary. The acting for this role was good, nothing great but bet
ter than Salem's.

The plot was real lousy if you think about it.
Stanley, who is presumed dead, makes his way onto the set of 'Cabin by the L
ake', the movie based on his script. He stumbles upon the director and in a
short time, the director is dead and Stanley is running the show. Yeah, out
of nowhere the crew is just going to let this stranger come into the picture
and finish the film not knowing anything about him. There's some killings, n
ot a whole lot, and the one's that are shown are ridiculous. One of the actr
esses on the set gets electrocuted while filming a scene. Another character
gets chewed up by a motorboat. And one gets tangled up in a plant before dro
wning. These writers must have been hard up for excitement.

I jus
t have to say that I was not impressed with the filming of the movie. The wa
y that it kept changing from looking low budget back to normal started to be
come irritating very fast. Also, the new cabin by the lake was poorly create
d. We aren't shown it but only in a few scenes, and the whole thing with the
chain in the basement was useless. It worked in the first film only because
we were shown the room a lot more, but it didn't work in this one.

There were too many characters in this sequel. All of them except for a fe
w had no reason to be there. The acting of what little is showed was really
bad and...they just didn't have a purpose in this movie.

All in a
ll, 'Return to Cabin by the Lake' is a sequel picking up from where the firs
t left off. 'Cabin by the Lake' I can take but this was just not impressive.
Judd Nelson should have avoided this one and so should you. It's nothing lik
e the first and it went entirely too slow. Nothing happened in the first hou
r and it continued to drag on for the second. Not to mention that the writin
g was horrible. Put this on only if you need some help getting to sleep.

So, we see that Stanley defies death and is still alive and well. By
the way he talks, it sounds like there could be a possible third installment
to a movie good just by itself. Quit throwing in sequels and we may be alrig
ht!

(Did the film makers not realize that they showed us how they
filmed the lake scenes from the first one? They were all done in a tank. Nev
er, never reveal the secrets of filming.)

Étiquette: Négative

Index de la note de review: 179
Texte de la note de review:
 Caught the tail end of this movie channel surfing through the cable movie c
hannels, and was so intrigued I sought out the next showing.

I re
ally didn't know what to expect after reading the program summary, but I cam
e away from this movie feeling quite disturbed and distressed. It also gave
me as adult who attended high school in the 80's, a little better insight in
to what our kids have to contend with these days.

The fact that y
ou don't see the shooting only adds to the chillness of the plot. To see bot
h child and adult alike struggle to comprehend and come to terms with the se
nseless shootings was at times overwhelming. And will admit that I shed quit
e a few tears throughout.

On the whole, not a movie that I would

seek out to what, however I am sure glad I did see it.
Étiquette: Positive

Index de la note de review: 179
Texte de la note de review:
 Caught the tail end of this movie channel surfing through the cable movie c
hannels, and was so intrigued I sought out the next showing.

I re
ally didn't know what to expect after reading the program summary, but I cam
e away from this movie feeling quite disturbed and distressed. It also gave
me as adult who attended high school in the 80's, a little better insight in
to what our kids have to contend with these days.

The fact that y
ou don't see the shooting only adds to the chillness of the plot. To see bot
h child and adult alike struggle to comprehend and come to terms with the se
nseless shootings was at times overwhelming. And will admit that I shed quit
e a few tears throughout.

On the whole, not a movie that I would
seek out to what, however I am sure glad I did see it.
Étiquette: Positive

Index de la note de review: 344
Texte de la note de review:
 Amateur, no budget films can be surprisingly good ... this however is not o
ne of them.

Ah, another Brad Sykes atrocity. The acting is hideou
s, except for Emmy Smith who shows some promise. The camera "direction" need
s serious reworking. And no more "hold the camera and run" gimmicks either;
it just doesn't work. The special effects are unimaginative, there's a probl
em when the effect can be identified in real time. If you're going to rip of
f an ear, please don't let us see the actor's real ear beneath the blood. Th
e scenery is bland and boring (same as Mr. Sykes other ventures), and the mu
sic is a cross between cheap motel porn and really bad guitar driven metal
(see the scenery comment).

Did I mention the lack of any real plo
t, or character development? Apparently, the scriptwriter didn't.
<br /
>Whoever is funding this guy ... please stop. I've seen some of his other "h
ome movies" (which I will not plug) and they are just as bad. Normally, a "d
irector" will grow and learn from his previous efforts ... not this guy. I
t's one thing to be an amateur filmmaker, but anyone can be a hack.

Definitely not even a popcorn film ... of course, chewing on popcorn kerne
ls would be less painful than this effort.

Award: The worst ever
military push-ups in a film.
Étiquette: Négative

Index de la note de review: 344
Texte de la note de review:
 Amateur, no budget films can be surprisingly good ... this however is not o
ne of them.

Ah, another Brad Sykes atrocity. The acting is hideou
s, except for Emmy Smith who shows some promise. The camera "direction" need
s serious reworking. And no more "hold the camera and run" gimmicks either;
it just doesn't work. The special effects are unimaginative, there's a probl
em when the effect can be identified in real time. If you're going to rip of
f an ear, please don't let us see the actor's real ear beneath the blood. Th
e scenery is bland and boring (same as Mr. Sykes other ventures), and the mu
sic is a cross between cheap motel porn and really bad guitar driven metal
(see the scenery comment).

Did I mention the lack of any real plo
t, or character development? Apparently, the scriptwriter didn't.
<br /
>Whoever is funding this guy ... please stop. I've seen some of his other "h
ome movies" (which I will not plug) and they are just as bad. Normally, a "d
irector" will grow and learn from his previous efforts ... not this guy. I

t's one thing to be an amateur filmmaker, but anyone can be a hack.

Definitely not even a popcorn film ... of course, chewing on popcorn kerne
ls would be less painful than this effort.

Award: The worst ever
military push-ups in a film.
Étiquette: Négative

Index de la note de review: 94
Texte de la note de review:
"Wisecracker," the biography of actor William Haines, offers a gratifying a
necdote about the former star when he was past 70 and long retired from maki
ng movies. The old gent was not sentimental and rarely watched his own film
s, but in 1972 he was persuaded to attend a Los Angeles museum screening of
SHOW PEOPLE, the late silent feature in which he co-starred with Marion Davi
es. Beforehand, Haines was worried that this comedy would provoke the wrong
kind of laughter, but he was pleasantly surprised (and no doubt relieved) at
how well it held up and how much the young audience enjoyed it. Watch the fi
lm today and you can see why: SHOW PEOPLE is a delightful Hollywood satire t
hat retains its charm because it lampoons its targets with wit and flair, ye
t without malice. It's still funny and its satirical points still resonate.
Needless to say, the technology of movie-making has changed vastly since the
silent days, but the pretensions and follies of the filmmakers themselves ha
ven't changed all that much.

SHOW PEOPLE also stands as the best
surviving work of Marion Davies, a first-rate comic performer who deserves a
prominent place in the pantheon of great comediennes. Where her career was c
oncerned Davies was both blessed and cursed by the patronage of her paramou
r, the newspaper magnate William Randolph Hearst. It's well known that Hears
t exerted enormous influence over Davies' choice of roles, and well known to
o that, despite her gift for comedy, he preferred to see her play dignified
heroines in period costume dramas. By the late '20s, for whatever reason, Ma
rion was permitted to strut her stuff in several exuberant light comedies (i
ncluding THE RED MILL and THE PATSY), but SHOW PEOPLE, directed by the great
King Vidor, stands as her most enjoyable showcase. William Haines gives an e
ngaging, likable performance as her boyfriend and co-star Billy Boone, but t
his is the leading lady's show all the way.

Marion plays Southern
belle Peggy Pepper, an aspiring actress who storms Hollywood accompanied by
her father, determined to become a movie star. (Her dad Colonel Pepper is pl
ayed by actor/director Dell Henderson, a veteran of Griffith's Biograph dram
as who■coincidentally?■resembled Hearst!) One of Marion's funniest bits, ofte
n excerpted elsewhere, is her audition at the Comet Studio casting office. W
hile Dad helpfully suggests emotions to portray ("Sorrow! . . . Joy!") and d
rops a handkerchief across her face, Peggy assumes the appropriate expressio
n and posture. She's hired, only to discover that Comet makes low-brow comed
ies, the kind of comedies where people squirt each other with seltzer and in
ept cops tumble over each other racing to the rescue. Of course, Comet is in
tended as a take-off of Mack Sennett's Keystone, but the true nature of the
satire becomes clear as the story unfolds. As Peggy Pepper rises in the movi
e star hierarchy she leaves Comet for the more prestigious High Art Studio,
assuming the name "Patricia Peppoire" as more befitting her new station in l
ife as a serious actress. At some point it may occur to us (as it surely did
to viewers in 1928) that Davies' rival Gloria Swanson started out in Keyston
e comedies before rising to prominence in serious dramas for Cecil B. DeMill
e. And as Miss Peppoire takes herself more and more seriously, giving the hi
gh-hat treatment to former colleagues such as lowly comic Billy Boone, Davie
s' performance takes on an element of wicked parody seemingly aimed squarely
at Swanson herself. This is especially notable during an interview sequence,
when Miss Peppoire's spokesman spouts pretentious nonsense while the star de

livers a spot-on impersonation of Swanson. I suppose this was intended as a
friendly spoof, but I have to wonder how friendly relations were between Glo
ria and Marion after this movie was released.

In any event, SHOW
PEOPLE is a delicious treat for buffs, who will relish the parade of star ca
meos throughout. Charlie Chaplin contributes a nice bit, sans makeup and loo
king quite distinguished, eagerly seeking Patricia Peppoire's autograph! (An
d in a show of good sportsmanship Marion Davies herself puts in a self-mocki
ng cameo appearance, evening the score for poking fun at Swanson by poking f
un at herself.) This is a silent film that viewers not especially attuned to
silents might appreciate, at least those viewers with a taste for movies abo
ut the movie business. SHOW PEOPLE surely belongs in the company of such cla
ssics as SUNSET BOULEVARD and SINGIN' IN THE RAIN, among Hollywood's most ex
pertly produced, enjoyable exercises in amused self-examination.
Étiquette: Positive

Index de la note de review: 94
Texte de la note de review:
"Wisecracker," the biography of actor William Haines, offers a gratifying a
necdote about the former star when he was past 70 and long retired from maki
ng movies. The old gent was not sentimental and rarely watched his own film
s, but in 1972 he was persuaded to attend a Los Angeles museum screening of
SHOW PEOPLE, the late silent feature in which he co-starred with Marion Davi
es. Beforehand, Haines was worried that this comedy would provoke the wrong
kind of laughter, but he was pleasantly surprised (and no doubt relieved) at
how well it held up and how much the young audience enjoyed it. Watch the fi
lm today and you can see why: SHOW PEOPLE is a delightful Hollywood satire t
hat retains its charm because it lampoons its targets with wit and flair, ye
t without malice. It's still funny and its satirical points still resonate.
Needless to say, the technology of movie-making has changed vastly since the
silent days, but the pretensions and follies of the filmmakers themselves ha
ven't changed all that much.

SHOW PEOPLE also stands as the best
surviving work of Marion Davies, a first-rate comic performer who deserves a
prominent place in the pantheon of great comediennes. Where her career was c
oncerned Davies was both blessed and cursed by the patronage of her paramou
r, the newspaper magnate William Randolph Hearst. It's well known that Hears
t exerted enormous influence over Davies' choice of roles, and well known to
o that, despite her gift for comedy, he preferred to see her play dignified
heroines in period costume dramas. By the late '20s, for whatever reason, Ma
rion was permitted to strut her stuff in several exuberant light comedies (i
ncluding THE RED MILL and THE PATSY), but SHOW PEOPLE, directed by the great
King Vidor, stands as her most enjoyable showcase. William Haines gives an e
ngaging, likable performance as her boyfriend and co-star Billy Boone, but t
his is the leading lady's show all the way.

Marion plays Southern
belle Peggy Pepper, an aspiring actress who storms Hollywood accompanied by
her father, determined to become a movie star. (Her dad Colonel Pepper is pl
ayed by actor/director Dell Henderson, a veteran of Griffith's Biograph dram
as who■coincidentally?■resembled Hearst!) One of Marion's funniest bits, ofte
n excerpted elsewhere, is her audition at the Comet Studio casting office. W
hile Dad helpfully suggests emotions to portray ("Sorrow! . . . Joy!") and d
rops a handkerchief across her face, Peggy assumes the appropriate expressio
n and posture. She's hired, only to discover that Comet makes low-brow comed
ies, the kind of comedies where people squirt each other with seltzer and in
ept cops tumble over each other racing to the rescue. Of course, Comet is in
tended as a take-off of Mack Sennett's Keystone, but the true nature of the
satire becomes clear as the story unfolds. As Peggy Pepper rises in the movi
e star hierarchy she leaves Comet for the more prestigious High Art Studio,

assuming the name "Patricia Peppoire" as more befitting her new station in l
ife as a serious actress. At some point it may occur to us (as it surely did
to viewers in 1928) that Davies' rival Gloria Swanson started out in Keyston
e comedies before rising to prominence in serious dramas for Cecil B. DeMill
e. And as Miss Peppoire takes herself more and more seriously, giving the hi
gh-hat treatment to former colleagues such as lowly comic Billy Boone, Davie
s' performance takes on an element of wicked parody seemingly aimed squarely
at Swanson herself. This is especially notable during an interview sequence,
when Miss Peppoire's spokesman spouts pretentious nonsense while the star de
livers a spot-on impersonation of Swanson. I suppose this was intended as a
friendly spoof, but I have to wonder how friendly relations were between Glo
ria and Marion after this movie was released.

In any event, SHOW
PEOPLE is a delicious treat for buffs, who will relish the parade of star ca
meos throughout. Charlie Chaplin contributes a nice bit, sans makeup and loo
king quite distinguished, eagerly seeking Patricia Peppoire's autograph! (An
d in a show of good sportsmanship Marion Davies herself puts in a self-mocki
ng cameo appearance, evening the score for poking fun at Swanson by poking f
un at herself.) This is a silent film that viewers not especially attuned to
silents might appreciate, at least those viewers with a taste for movies abo
ut the movie business. SHOW PEOPLE surely belongs in the company of such cla
ssics as SUNSET BOULEVARD and SINGIN' IN THE RAIN, among Hollywood's most ex
pertly produced, enjoyable exercises in amused self-examination.
Étiquette: Positive

Index de la note de review: 524
Texte de la note de review:
:Spoilers:

I was very disappointed in Love's Abiding Joy. I had
been waiting a really long time to see it and I finally got the chance when
it re-aired Thursday night on Hallmark. I love the first three "Love" movies
but this one was nothing like I thought it was going to be. The whole movie
was sad and depressing, there were way to many goofs, and the editing was ve
ry poor - to many scenes out of context. I also think the death of baby Kath
y happened way to soon and Clarks appearance in the movie just didn't seem t
o fit. It seemed like none of the actors really wanted to be there - they we
re all lacking emotion. There seemed to be no interaction between Missie and
Willie at all.

I think the script writers should have went more b
y the book. It seems like every movie that's been made so far just slips fur
ther and further away from Janette Oke's writings. I mean in the movie they
never mentioned a thing about the mine and the two boys or Clark getting hur
t because of it. And I think Missie and Willies reactions to Kathy's death c
ould have been shown and heard rather than just heard.

Out of the
four movies that have been made so far I'd have to say that Love's Abiding J
oy is my least favorite. I hope with the next four movies that more of the b
ook is followed and if Clarks character is in them I hope he's got a bigger
part and I hope his part isn't so bland. I also hope there is more of Scotti
e and Cookie and maybe even Marty but who knows what the script writers will
have in store next.
Étiquette: Négative

Index de la note de review: 524
Texte de la note de review:
:Spoilers:

I was very disappointed in Love's Abiding Joy. I had
been waiting a really long time to see it and I finally got the chance when
it re-aired Thursday night on Hallmark. I love the first three "Love" movies
but this one was nothing like I thought it was going to be. The whole movie
was sad and depressing, there were way to many goofs, and the editing was ve

```
ry poor - to many scenes out of context. I also think the death of baby Kath
y happened way to soon and Clarks appearance in the movie just didn't seem t
o fit. It seemed like none of the actors really wanted to be there - they we
re all lacking emotion. There seemed to be no interaction between Missie and
Willie at all.<br /><br />I think the script writers should have went more b
y the book. It seems like every movie that's been made so far just slips fur
ther and further away from Janette Oke's writings. I mean in the movie they
never mentioned a thing about the mine and the two boys or Clark getting hur
t because of it. And I think Missie and Willies reactions to Kathy's death c
ould have been shown and heard rather than just heard.<br /><br />Out of the
four movies that have been made so far I'd have to say that Love's Abiding J
oy is my least favorite. I hope with the next four movies that more of the b
ook is followed and if Clarks character is in them I hope he's got a bigger
part and I hope his part isn't so bland. I also hope there is more of Scotti
e and Cookie and maybe even Marty but who knows what the script writers will
have in store next.
Étiquette: Négative
```

Vectorisation

In [9]:
```python
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
```

In [10]:
```python
def vectorize_text(data, vectorizer):
        """Vectorize text data using specified vectorizer."""
        vectorized_data = vectorizer.fit_transform(data)
        return vectorized_data, vectorizer
```

In [11]:
```python
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_features=5000)
```

In [12]:
```python
count_vectorizer = CountVectorizer(stop_words='english', max_features=5000)
```

In [13]:
```python
features, current_vectorizer = vectorize_text(reviews_df['review'], tfidf_ve
print("Forme de la matrice TF-IDF :", features.shape)
```

```
Forme de la matrice TF-IDF : (602, 5000)
```

In [14]:
```python
features_count, _ = vectorize_text(reviews_df['review'], count_vectorizer)
print("Forme de la matrice Count :", features_count.shape)
```

```
Forme de la matrice Count : (602, 5000)
```

Entraînement

1. Logistic Regression

In [15]:
```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score
```

In [16]:
```python
def train_and_evaluate(features, labels):
        """Train and evaluate a logistic regression model."""
        X_train, X_test, y_train, y_test = train_test_split(features, labels
```

```python
        model = LogisticRegression(random_state=42)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        print("Accuracy:", accuracy_score(y_test, y_pred))
        print("Classification Report:\n", classification_report(y_test, y_pr
```

In [17]:
```python
train_and_evaluate(features, reviews_df['label'])
```

```
Accuracy: 0.8543046357615894
Classification Report:
               precision    recall  f1-score   support

           0       0.81      0.88      0.85        68
           1       0.90      0.83      0.86        83

    accuracy                           0.85       151
   macro avg       0.85      0.86      0.85       151
weighted avg       0.86      0.85      0.85       151
```

In [18]:
```python
train_and_evaluate(features_count, reviews_df['label'])
```

```
Accuracy: 0.8079470198675497
Classification Report:
               precision    recall  f1-score   support

           0       0.79      0.78      0.79        68
           1       0.82      0.83      0.83        83

    accuracy                           0.81       151
   macro avg       0.81      0.81      0.81       151
weighted avg       0.81      0.81      0.81       151
```

2. SVM & GridSearchCV

In [19]:
```python
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
```

In [20]:
```python
def train_and_evaluate_svm(X_train, y_train, X_test, y_test):
        param_grid = {
                'C': [0.1, 1, 10, 100],
                'kernel': ['linear', 'rbf'],
                'gamma': ['scale', 'auto']
        }
        grid_search = GridSearchCV(SVC(), param_grid, cv=5, scoring='accurac
        grid_search.fit(X_train, y_train)
        print("Best parameters:", grid_search.best_params_)
        best_model = grid_search.best_estimator_
        y_pred = best_model.predict(X_test)
        print("Accuracy:", accuracy_score(y_test, y_pred))
        print("Classification Report:\n", classification_report(y_test, y_pr
```

In [21]:
```python
train_and_evaluate_svm(features, reviews_df['label'], features, reviews_df['
```

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits
Best parameters: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
Accuracy: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00       301
           1       1.00      1.00      1.00       301

    accuracy                           1.00       602
   macro avg       1.00      1.00      1.00       602
weighted avg       1.00      1.00      1.00       602
```

In [22]:
```python
train_and_evaluate_svm(features_count, reviews_df['label'], features_count,
```

```
Fitting 5 folds for each of 16 candidates, totalling 80 fits
Best parameters: {'C': 100, 'gamma': 'auto', 'kernel': 'rbf'}
Accuracy: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00       301
           1       1.00      1.00      1.00       301

    accuracy                           1.00       602
   macro avg       1.00      1.00      1.00       602
weighted avg       1.00      1.00      1.00       602
```

3. Random Forest

In [23]:
```python
from sklearn.ensemble import RandomForestClassifier
```

In [24]:
```python
def train_and_evaluate_rf(features, labels):
        X_train, X_test, y_train, y_test = train_test_split(features, labels
        model = RandomForestClassifier(n_estimators=100, random_state=42)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        print("Accuracy:", accuracy_score(y_test, y_pred))
        print("Classification Report:\n", classification_report(y_test, y_pr
```

In [25]:
```python
train_and_evaluate_rf(features, reviews_df['label'])
```

```
Accuracy: 0.7549668874172185
Classification Report:
               precision    recall  f1-score   support

           0       0.70      0.81      0.75        68
           1       0.82      0.71      0.76        83

    accuracy                           0.75       151
   macro avg       0.76      0.76      0.75       151
weighted avg       0.76      0.75      0.76       151
```

In [26]:
```python
train_and_evaluate_rf(features_count, reviews_df['label'])
```

```
Accuracy: 0.7549668874172185
Classification Report:
               precision    recall  f1-score   support

           0       0.68      0.87      0.76        68
           1       0.86      0.66      0.75        83

    accuracy                           0.75       151
   macro avg       0.77      0.77      0.75       151
weighted avg       0.78      0.75      0.75       151
```

4. Naive Bayes

In [27]:
```python
from sklearn.naive_bayes import MultinomialNB
```

In [28]:
```python
def train_and_evaluate_nb(features, labels):
        X_train, X_test, y_train, y_test = train_test_split(features, labels
        model = MultinomialNB()
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        print("Accuracy:", accuracy_score(y_test, y_pred))
        print("Classification Report:\n", classification_report(y_test, y_pr
```

In [29]:
```python
train_and_evaluate_nb(features, reviews_df['label'])
```

```
Accuracy: 0.8145695364238411
Classification Report:
               precision    recall  f1-score   support

           0       0.74      0.91      0.82        68
           1       0.91      0.73      0.81        83

    accuracy                           0.81       151
   macro avg       0.82      0.82      0.81       151
weighted avg       0.83      0.81      0.81       151
```

In [30]:
```python
train_and_evaluate_nb(features_count, reviews_df['label'])
```

```
Accuracy: 0.8410596026490066
Classification Report:
               precision    recall  f1-score   support

           0       0.78      0.91      0.84        68
           1       0.92      0.78      0.84        83

    accuracy                           0.84       151
   macro avg       0.85      0.85      0.84       151
weighted avg       0.85      0.84      0.84       151
```

5. Decision Tree

In [31]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [32]:
```
def train_and_evaluate_dt(features, labels):
        X_train, X_test, y_train, y_test = train_test_split(features, labels
        model = DecisionTreeClassifier(random_state=42)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        print("Accuracy:", accuracy_score(y_test, y_pred))
        print("Classification Report:\n", classification_report(y_test, y_pr
```

In [33]:
```
train_and_evaluate_dt(features, reviews_df['label'])
```

```
Accuracy: 0.6622516556291391
Classification Report:
               precision    recall  f1-score   support

           0       0.62      0.66      0.64        68
           1       0.71      0.66      0.68        83

    accuracy                           0.66       151
   macro avg       0.66      0.66      0.66       151
weighted avg       0.67      0.66      0.66       151
```

In [34]:
```
train_and_evaluate_dt(features_count, reviews_df['label'])
```

```
Accuracy: 0.6291390728476821
Classification Report:
               precision    recall  f1-score   support

           0       0.58      0.63      0.61        68
           1       0.68      0.63      0.65        83

    accuracy                           0.63       151
   macro avg       0.63      0.63      0.63       151
weighted avg       0.63      0.63      0.63       151
```

6. K-Nearest Neighbors

In [35]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [36]:
```
def train_and_evaluate_knn(features, labels):
        X_train, X_test, y_train, y_test = train_test_split(features, labels
        model = KNeighborsClassifier()
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        print("Accuracy:", accuracy_score(y_test, y_pred))
        print("Classification Report:\n", classification_report(y_test, y_pr
```

In [37]:
```
train_and_evaluate_knn(features, reviews_df['label'])
```

```
Accuracy: 0.8675496688741722
Classification Report:
               precision    recall  f1-score   support

           0       0.85      0.85      0.85        68
           1       0.88      0.88      0.88        83

    accuracy                           0.87       151
   macro avg       0.87      0.87      0.87       151
weighted avg       0.87      0.87      0.87       151
```

In [38]:
```python
train_and_evaluate_knn(features_count, reviews_df['label'])
```

```
Accuracy: 0.6291390728476821
Classification Report:
               precision    recall  f1-score   support

           0       0.60      0.53      0.56        68
           1       0.65      0.71      0.68        83

    accuracy                           0.63       151
   macro avg       0.62      0.62      0.62       151
weighted avg       0.63      0.63      0.63       151
```

7. XGBoost

In [39]:
```python
from xgboost import XGBClassifier
```

In [40]:
```python
def train_and_evaluate_xgb(features, labels):
        X_train, X_test, y_train, y_test = train_test_split(features, labels
        model = XGBClassifier()
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        print("Accuracy:", accuracy_score(y_test, y_pred))
        print("Classification Report:\n", classification_report(y_test, y_pr
```

In [41]:
```python
train_and_evaluate_xgb(features, reviews_df['label'])
```

```
Accuracy: 0.7417218543046358
Classification Report:
               precision    recall  f1-score   support

           0       0.71      0.72      0.72        68
           1       0.77      0.76      0.76        83

    accuracy                           0.74       151
   macro avg       0.74      0.74      0.74       151
weighted avg       0.74      0.74      0.74       151
```

In [42]:
```python
train_and_evaluate_xgb(features_count, reviews_df['label'])
```

```
Accuracy: 0.7483443708609272
Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.76      0.73        68
           1       0.79      0.73      0.76        83

    accuracy                           0.75       151
   macro avg       0.75      0.75      0.75       151
weighted avg       0.75      0.75      0.75       151
```

Evaluation

In [43]:
```
comparison_data = {
    "Model": [],
    "Vectorization": [],
    "Accuracy": [],
    "Precision (macro)": [],
    "Recall (macro)": [],
    "F1-score (macro)": [],
}
```

In [44]:
```
def add_to_comparison(model_name, vectorization, accuracy, report):
        """Add data to the comparison table."""
        comparison_data["Model"].append(model_name)
        comparison_data["Vectorization"].append(vectorization)
        comparison_data["Accuracy"].append(accuracy)
        comparison_data["Precision (macro)"].append(report["macro avg"]["pre
        comparison_data["Recall (macro)"].append(report["macro avg"]["recall
        comparison_data["F1-score (macro)"].append(report["macro avg"]["f1-s
```

In [45]:
```
def run_and_add_to_comparison(features, labels, model_name, model, vectoriza
        """Run the model and add it to the comparison table."""
        X_train, X_test, y_train, y_test = train_test_split(features, labels
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
        report = classification_report(y_test, y_pred, output_dict=True)
        add_to_comparison(model_name, vectorization_type, accuracy, report)
```

In [46]:
```
run_and_add_to_comparison(features, reviews_df['label'], "Logistic Regressic
run_and_add_to_comparison(features_count, reviews_df['label'], "Logistic Reg

run_and_add_to_comparison(features, reviews_df['label'], "Random Forest", Ra
run_and_add_to_comparison(features_count, reviews_df['label'], "Random Fores

run_and_add_to_comparison(features, reviews_df['label'], "Naive Bayes", Mult
run_and_add_to_comparison(features_count, reviews_df['label'], "Naive Bayes"

run_and_add_to_comparison(features, reviews_df['label'], "SVM", SVC(), "TF-I
run_and_add_to_comparison(features_count, reviews_df['label'], "SVM", SVC(),

run_and_add_to_comparison(features, reviews_df['label'], "Decision Tree", De
```

```
run_and_add_to_comparison(features_count, reviews_df['label'], "Decision Tre

run_and_add_to_comparison(features, reviews_df['label'], "KNN", KNeighborsCl
run_and_add_to_comparison(features_count, reviews_df['label'], "KNN", KNeigh

run_and_add_to_comparison(features, reviews_df['label'], "XGBoost", XGBClass
run_and_add_to_comparison(features_count, reviews_df['label'], "XGBoost", XG
```

In [47]:

```
comparison_df = pd.DataFrame(comparison_data)
comparison_df.head(20)
```

Out[47]:

	Model	Vectorization	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)
0	Logistic Regression	TF-IDF	0.854305	0.853457	0.856839	0.853785
1	Logistic Regression	Count	0.807947	0.806237	0.805369	0.805766
2	Random Forest	TF-IDF	0.754967	0.757823	0.759833	0.754795
3	Random Forest	Count	0.754967	0.768768	0.765149	0.754795
4	Naive Bayes	TF-IDF	0.814570	0.824271	0.823352	0.814561
5	Naive Bayes	Count	0.841060	0.845246	0.847449	0.840997
6	SVM	TF-IDF	0.854305	0.856340	0.859497	0.854145
7	SVM	Count	0.761589	0.762454	0.753898	0.755927
8	Decision Tree	TF-IDF	0.662252	0.660783	0.662208	0.660764
9	Decision Tree	Count	0.629139	0.628203	0.629429	0.627817
10	KNN	TF-IDF	0.867550	0.866230	0.866230	0.866230
11	KNN	Count	0.629139	0.624176	0.620128	0.620330
12	XGBoost	TF-IDF	0.741722	0.739219	0.739812	0.739482
13	XGBoost	Count	0.748344	0.747455	0.749823	0.747447

In [ ]: