Recomendador de películas y análisis de sentimiento

IMDb y Twitter

Ela Katherine Shepherd Arévalo
Pablo Daurell Marina

Índice

1 Análisis de sentimiento

- Dataset
- Naive Bayes
- Árbol de decisión
- Red Neuronal

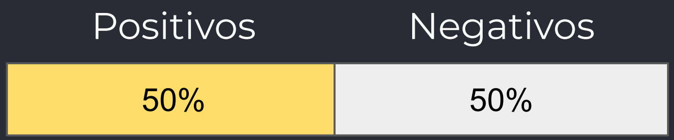
2 Recomendador de películas

- Dataset
- K-NN
- Tweepy
- Recomendador + Sentimiento

Análisis de sentimiento

Datos

- Dataset: [Sentiment140]
  - 1.600.000 Tweets (inglés) clasificados
    (Negativo 0, Positivo 4)



  - Columnas: *sentiment, id, date, query, user, text*

Datos

pandas

	sentiment	text
0	0	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	is upset that he can't update his Facebook by ...
2	0	@Kenichan I dived many times for the ball. Man...
3	0	my whole body feels itchy and like its on fire
4	0	@nationwideclass no, it's not behaving at all....
...	...	...
1599995	1	Just woke up. Having no school is the best fee...
1599996	1	TheWDB.com - Very cool to hear old Walt interv...
1599997	1	Are you ready for your MoJo Makeover? Ask me f...
1599998	1	Happy 38th Birthday to my boo of alll time!!! ...
1599999	1	happy #charitytuesday @theNSPCC @SparksCharity...

- Preprocesado:
  - Positivo: 4 → 1
  - Sólo columnas: *sentiment, text*
  - Texto en minúsculas
  - Eliminar enlaces y menciones *(@user)*

Técnicas utilizadas

- Naive Bayes
  - Multinomial NB
  - Bernoulli NB
  - Complement NB
- Árbol de decisión
- Red Neuronal

NLTK

scikit learn

TensorFlow

Datos

- Naive-Bayes y Árbol Decisión:
    - **70%** train, **30%** test
    - **Stopwords**: Eliminar “no”, “not”, “nor”
- Red Neuronal:
    - **70%** train, **15%** validation, **15%** test

Naive-Bayes

Árbol Decisión

Naive Bayes y Árbol Decisión

Datos de entrada

Bolsa de palabras:

N-gramas:

- Monogramas
- Bigramas
- **Bi + monogramas**

	(1,1)	**(1,2)**	(2,2)
Train	80.18%	**91.73%**	93.11%
Test	77.18%	**78.30%**	72.67%

Preprocesamiento adicional:

- Lemmatization
- **Stemming**

	Lem.	**Stem.**
Train	91.73%	**91.75%**
Test	78.30%	**78.31%**

Naive Bayes y Árbol Decisión

Datos de entrada

Bolsa de palabras:

Tipo de NB:

- Complement
- Bernoulli
- **Multinomial**
- ~~(Gauss)~~

	Complement	Bernoulli	**Multinomial**
Train	91.75%	91.71%	**91.75%**
Test	78.31%	78.5%	**78.31%**

Your session crashed after using all available RAM. View runtime logs

Naive Bayes y Árbol Decisión

Resultados

TF-IDF:

Usamos TfidfTransformer sobre CountVectorizer

Train	90.82%
Test	78.46%

```
tfidfer = TfidfTransformer()
tf_vector_traindata = tfidfer.fit_transform(cv_vector_traindata_stem)

tf_mnb_classifier = MultinomialNB()
tf_mnb_classifier.fit(tf_vector_traindata, train_label)

tf_vector_testdata = tfidfer.transform(cv_vector_testdata_stem)
tf_predictions_train = tf_mnb_classifier.predict(tf_vector_traindata)
tf_predictions_test = tf_mnb_classifier.predict(tf_vector_testdata)

print("Multinomial Naive Bayes, tf-idf, train success rate:", np.mean(tf_predictions_train == train_label)*100, "%")
print("Multinomial Naive Bayes, tf-idf, test success rate:", np.mean(tf_predictions_test == test_label)*100, "%")
```

```
Multinomial Naive Bayes, tf-idf, train success rate: 90.81678571428571 %
Multinomial Naive Bayes, tf-idf, test success rate: 78.45854166666668 %
```

Mejora: menos “overfitting”

Naive Bayes y Árbol Decisión

Resultados

Árbol decisión:


```
tree_train_acc = []
tree_test_acc = []
max_dep = np.arange(0, 30, 5)
for i in max_dep[1:]:
  tree_classifier = DecisionTreeClassifier(criterion="gini", max_depth=i)
  tree_classifier.fit(tf_vector_traindata, train_label)
  tree_predictions_train = tree_classifier.predict(tf_vector_traindata)
  tree_predictions_test = tree_classifier.predict(tf_vector_testdata)
  tree_train_acc.append(np.mean(tree_predictions_train == train_label))
  tree_test_acc.append(np.mean(tree_predictions_test == test_label))
```

Red Neuronal

Red Neuronal

Datos de entrada

TextVectorization:

- Estandarizar cada tweet
- Tweet → Vector numérico

```
max_features = 10000
sequence_length = 140

vectorize_layer = TextVectorization(
    standardize=custom_standardization,
    max_tokens=max_features,
    output_mode='int',
    output_sequence_length=sequence_length)
```

Red Neuronal

Modelo

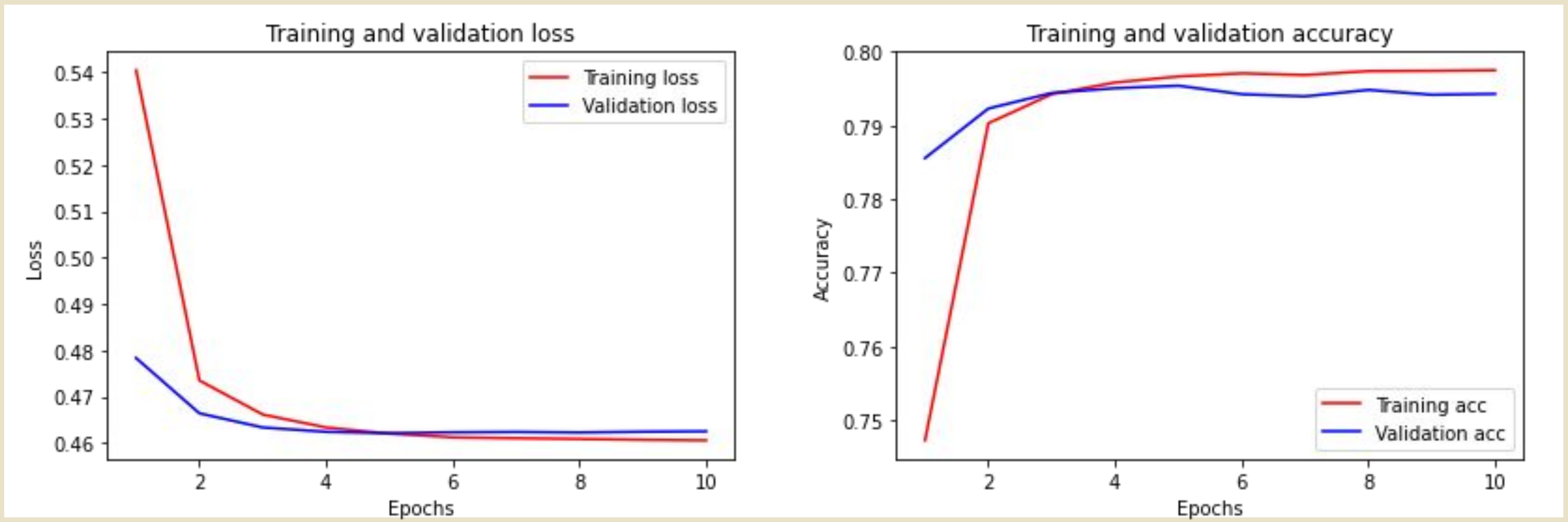
Capas:

- **Embedding:** Tweets vectorizados → Word-Embedding
- **Dropout:** Regularización (evitar sobreaprendizaje)
- **GlobalAveragePooling1D:** Entrada → Vector tamaño fijo
- **Dense:** Realizar clasificación

```
model = tf.keras.Sequential([
    layers.Embedding(max_features + 1, embedding_dim),
    layers.Dropout(0.2),
    layers.GlobalAveragePooling1D(),
    layers.Dropout(0.2),
    layers.Dense(1)
])
```

Red Neuronal

Resultados

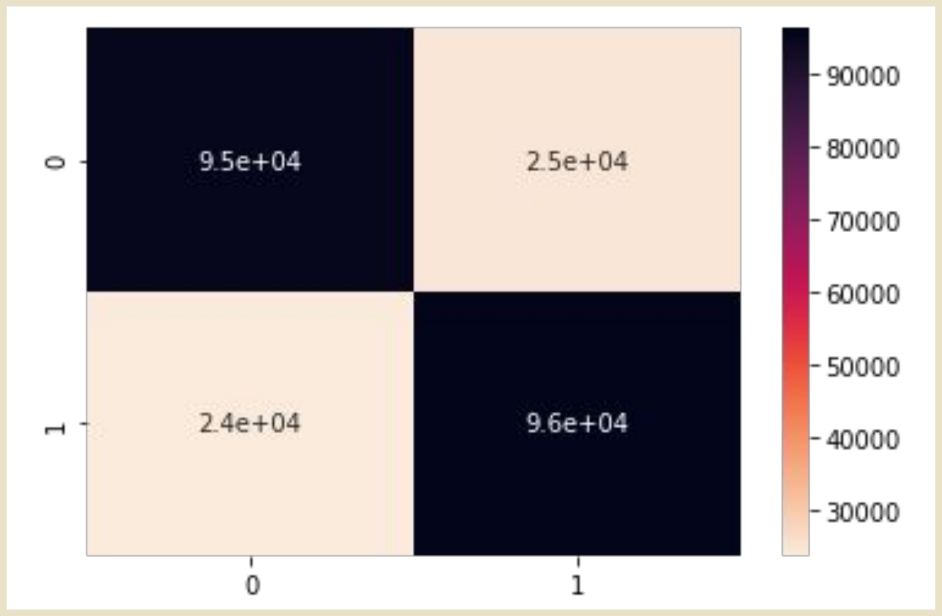



```
Train loss: 45.31547129154205%, Test loss: 46.077895164489746%
Train accuracy: 80.0108015537262%, Test accuracy: 79.5870840549469%
```

Red Neuronal

Resultados

Test Data



	Test
Accuracy (Correct predictions)	79.58%
Precission (Correct predicted positives)	78%
Recall (Real Positives Captured)	82.3%
F1	80%

Recomendador de películas

IMDb

Collaborative Filtering:

- En función de los <u>géneros más vistos</u> y de <u>usuarios similares</u>

Algoritmo K-NN

Datos

Dataset: MovieTweetings

Users

Tweets
de cada usuario

(70550, 2)

	user id	twitter id
0	1	139564917
1	2	17528189
2	3	522540374
3	4	475571186
4	5	215022153

Movies

Géneros
más vistos por cada usuario

(37248, 3)

	movie id	movie title	genre
0	0000008	Edison Kinetoscopic Record of a Sneeze (1894)	Documentary|Short
1	0000010	La sortie des usines Lumière (1895)	Documentary|Short
2	0000012	The Arrival of a Train (1896)	Documentary|Short
3	25	The Oxford and Cambridge University Boat Race ...	NaN
4	0000091	Le manoir du diable (1896)	Short|Horror

Ratings

Películas
vistas por cada usuario

(903946, 4)

	user id	movie id	rating	rating timestamp
0	1	0114508	8	1381006850
1	2	0499549	9	1376753198
2	2	1305591	8	1376742507
3	2	1428538	1	1371307089
4	3	0075314	1	1595468524

Algoritmo K-NN

Datos de entrada

- Modificar dataset original:
  - Usuario: Vector con cantidad de géneros vistos

- <u>Objetivo</u>: Calcular distancias entre “usuarios” y buscar más cercanos

	user id	Documentary	Short	Horror	Comedy	Action	Adventure	Fantasy	Sci-Fi	Crime	Western	Drama	Romance	History	Family	War	Sport	Biography	Mystery	Thriller	Animation	Music	Musical	Film-Noir	Adult	Talk-Show	News	Reality-TV	Game-Show
1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2	2	0	0	1	0	2	2	2	2	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	3	0	0	3	6	3	3	3	1	11	0	15	1	0	0	1	0	2	4	8	0	1	0	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0	0	1	0	5	2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
5	5	0	0	0	3	0	1	0	0	1	0	5	2	2	0	2	0	2	0	2	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
70545	70545	0	0	50	17	26	31	19	34	11	1	40	5	0	9	0	0	2	37	65	4	1	1	0	0	0	0	0	0
70546	70546	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
70547	70547	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70548	70548	0	0	8	24	23	11	7	6	17	2	52	8	5	5	6	2	10	11	27	1	0	2	0	0	0	0	0	0
70549	70549	0	0	0	0	1	1	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

70549 rows × 29 columns

Algoritmo K-NN

Ejemplo

```
uid = 3
li = classifier.kneighbors([X[uid-1]],n_neighbors=5,return_distance=False)
li = np.delete(li[0], 0) # Eliminamos al propio usuario
li = list(map(lambda x : x + 1, li)) # Sumar 1 para hacer coincidir los indices de la tabla con el user_id
```

```
print('Usuarios similares al usuario nº{}: {}'.format(uid, li))
```

```
Usuarios similares al usuario nº3: [6101, 2616, 47348, 6063]
```

	user id	Documentary	Short	Horror	Comedy	Action	Adventure	Fantasy	Sci-Fi	Crime	Western	Drama	Romance	History	Family	War	Sport	Biography	Mystery	Thriller	Animation	Music	Musical	Film-Noir	Adult	Talk-Show	News	Reality-TV	Game-Show
3	3	0	0	3	6	3	3	3	1	11	0	15	1	0	0	1	0	2	4	8	0	1	0	0	0	0	0	0	0

	user id	Documentary	Short	Horror	Comedy	Action	Adventure	Fantasy	Sci-Fi	Crime	Western	Drama	Romance	History	Family	War	Sport	Biography	Mystery	Thriller	Animation	Music	Musical	Film-Noir	Adult	Talk-Show	News	Reality-TV	Game-Show
6101	6101	0	0	3	6	3	2	1	1	7	0	13	1	0	0	0	0	2	5	10	0	1	0	0	0	0	0	0	0

Algoritmo K-NN

Ejemplo

```
Ids peliculas vistas por usuario 3:
['0075314' '0102926' '0114369' '0118715' '0120737' '0208092' '0358273'
 '0477348' '10039344' '1051906' '1568346' '2278388' '6199572' '6723592'
 '6751668' '7131622' '7975244' '7984734' '8367814' '8579674' '8946378']

Ids peliculas vistas por usuario similar (6101):
['0118971' '0154506' '0166924' '0256380' '1038988' '1206543' '1229340'
 '1242422' '1291150' '1294970' '1535109' '1614989' '1800241' '1939659'
 '2106476' '2140373' '2193215' '2226417' '2278388']
```

```
Peliculas a recomendar:
['0118971' '0154506' '0166924' '0256380' '1038988' '1206543' '1229340'
 '1242422' '1291150' '1294970' '1535109' '1614989' '1800241' '1939659'
 '2106476' '2140373' '2193215' '2226417']
```

```
Titulos (y géneros) de las peliculas recomendadas:

The Devil's Advocate (1997):    (Drama|Mystery|Thriller)
Following (1998):       (Crime|Mystery|Thriller)
Mulholland Dr. (2001):  (Drama|Mystery|Thriller)
Shallow Hal (2001):     (Comedy|Drama|Fantasy|Romance)
[Rec] (2007):   (Horror|Mystery|Thriller)
Out of the Furnace (2013):      (Crime|Drama|Thriller)
Anchorman 2: The Legend Continues (2013):       (Comedy)
Celda 211 (2009):       (Action|Crime|Drama|Thriller)
Teenage Mutant Ninja Turtles (2014):    (Action|Adventure|Comedy|Sci-Fi)
The Angriest Man in Brooklyn (2014):    (Comedy|Drama)
Captain Phillips (2013):        (Biography|Drama|Thriller)
Hodejegerne (2011):     (Action|Crime|Thriller)
American Hustle (2013):         (Crime|Drama)
Carrie (2013):  (Drama|Horror)
Jagten (2012):  (Drama)
Saving Mr. Banks (2013):        (Biography|Comedy|Drama|Music)
The Counselor (2013):   (Crime|Drama|Thriller)
Insidious: Chapter 2 (2013):    (Horror|Mystery|Thriller)
```

Análisis de sentimiento (Twitter)

- Extraer tweets:
  - ID usuario → ID Twitter
  - `user = api.get_user(int(get_twitterID(id)))`
  - `tweets = api.user_timeline(screen_name=user.screen_name, count= 100, include_rts=False)`

	user id	twitter id
0	1	139564917
1	2	17528189
2	3	522540374
3	4	475571186
4	5	215022153

Análisis de sentimiento (Twitter)

- Análisis sentimiento:
  - Red Neuronal

```
@LeeDawsonPT The duality of entertainment

@alanswan a receipt of all the non essential buys from amazon- mainly the harmonica and cocktail mixers. A lethal combo

Fanatastic campaign launched by SIRO. Life is different now and we're all adjusting as best we can! #StayHome https://t.co/eeuZ0Ig0ai

@hoeyannie @labour Congrats!!

@NianticHelp What about the special research tasks with time limits? Hard to get out of the home to watch very spec… https://t.co/NgIyNKZWAI

@LeeDawsonPT go to town with it https://t.co/j4B9L8jdi1

@Ghost_Fl0wer Hope that it turns out you don't have OC. Here if you need anything

@LfcHarsh Repetitive music boosts productivity for both so that's prolly why!

Brave and bold statement from British TV icon Phillip Schofield, nothing but support to him and his family should b… https://t.co/wbhTnb2oMj

I've had people call me all sorts of things but I think my favourite has to be the person who thought "Airline" was… https://t.co/PoQUb0Ad85
```

```
User id: 1
Number of tweets: 10
- User name: @Waffaboy
- User sentiment: Positive
```

Recomendador + Sentimiento

Enfoque **principal**:

- Género → Sentimiento
  - (1 pos, 0 neutro, -1 neg)

- Película → ∑ géneros
  - (> 0 pos, == 0 neutro, < 0 neg)

'Documentary': 0, **'Short'**: 0, **'Horror'**: -1, **'Comedy'**: 1, **'Action'**: 1, **'Adventure'**: 1 ,**'Fantasy'**: 1, **'Sci-Fi'**: 1, **'Crime'**: -1, **'Western'**: 1, **'Drama'**: -1, **'Romance'**: 1, **'History'**: 0, **'Family'**: 1, **'War'**: -1, **'Sport'**: 1, **'Biography'**: 0, **'Mystery'**: -1, **'Thriller'**: -1, **'Animation'**: 0, **'Music'**: 1, **'Musical'**: 1, **'Film-Noir'**: -1, **'Adult'**: -1, **'Talk-Show'**: 1, **'News'**: 0, **'Reality-TV'**: 1, **'Game-Show'**: 1

Enfoque “**experimental**”:

- Película: Extraer sinopsis (IMDBpy)
  - `movie = ia.get_movie(id)`
  - `plot = movie.get('plot')`

- Sinopsis + Red Neuronal → Sentimiento
  - RN entrenada con tweets
  - Posible mejora: entrenar con películas

Recomendador + Sentimiento

Enfoque **principal**:

```
- User name: @theashoxford
- User sentiment: Positive
- Recomendations by similar users:
        - Cast Away (2000) : ( Adventure|Drama|Romance ):  1
        - The Irishman (2019) : ( Biography|Crime|Drama|History|Thriller ):  -1
        - Once Upon a Time ...in Hollywood (2019) : ( Comedy|Drama ):  0
        - Marriage Story (2019) : ( Comedy|Drama ):  0
        - The King (2019) : ( Biography|Drama|History|Romance|War ):  -1
        - The Two Popes (2019) : ( Biography|Comedy|Drama ):  0
        - A Perfect World (1993) : ( Crime|Drama|Thriller ):  -1
        - Jersey Girl (2004) : ( Comedy|Drama|Romance ):  1
        - My One and Only (2009) : ( Adventure|Biography|Comedy|Drama|Romance ):  1
        - Macbeth (2015) : ( Drama|History|War ):  -1
        - Amadeus (1984) : ( Biography|Drama|History|Music ):  0
        - Ard al-Khof (1999) : ( Thriller ):  -1
        - Hotel Rwanda (2004) : ( Biography|Drama|History|War ):  -1
        - Eat Pray Love (2010) : ( Drama|Romance ):  0
        - American Hustle (2013) : ( Crime|Drama ):  -1
        - Birdman (2014) : ( Comedy|Drama ):  0
        - Me and Earl and the Dying Girl (2015) : ( Comedy|Drama|Romance ):  1
        - Persepolis (2007) : ( Animation|Biography|Drama|History|War ):  -1
        - Me Before You (2016) : ( Drama|Romance ):  0
        - The Dressmaker (2015) : ( Comedy|Drama ):  0
        - A Hologram for the King (2016) : ( Comedy|Drama|Romance ):  1
        - Septembers of Shiraz (2015) : ( Thriller ):  -1
        - Lady Bird (2017) : ( Comedy|Drama ):  0
```

Enfoque “**experimental**”:

```
- User name: @theashoxford
- User sentiment: Positive
- Recomendations by similar users:
        - Cast Away (2000) : ( Adventure|Drama|Romance ):  1
        - The Irishman (2019) : ( Biography|Crime|Drama|History|Thriller ):  1
        - Once Upon a Time ...in Hollywood (2019) : ( Comedy|Drama ):  1
        - Marriage Story (2019) : ( Comedy|Drama ):  1
        - The King (2019) : ( Biography|Drama|History|Romance|War ):  -1
        - The Two Popes (2019) : ( Biography|Comedy|Drama ):  1
        - A Perfect World (1993) : ( Crime|Drama|Thriller ):  -1
        - Jersey Girl (2004) : ( Comedy|Drama|Romance ):  1
        - My One and Only (2009) : ( Adventure|Biography|Comedy|Drama|Romance ):  -1
        - Macbeth (2015) : ( Drama|History|War ):  1
        - Amadeus (1984) : ( Biography|Drama|History|Music ):  -1
        - Ard al-Khof (1999) : ( Thriller ):  -1
        - Hotel Rwanda (2004) : ( Biography|Drama|History|War ):  -1
        - Eat Pray Love (2010) : ( Drama|Romance ):  -1
        - American Hustle (2013) : ( Crime|Drama ):  1
        - Birdman (2014) : ( Comedy|Drama ):  1
        - Me and Earl and the Dying Girl (2015) : ( Comedy|Drama|Romance ):  -1
        - Persepolis (2007) : ( Animation|Biography|Drama|History|War ):  1
        - Me Before You (2016) : ( Drama|Romance ):  1
        - The Dressmaker (2015) : ( Comedy|Drama ):  1
        - A Hologram for the King (2016) : ( Comedy|Drama|Romance ):  -1
        - Septembers of Shiraz (2015) : ( Thriller ):  -1
        - Lady Bird (2017) : ( Comedy|Drama ):  1
```

Recomendador + Sentimiento

A story about a **police** officer who was assigned to a secret mission as an undercover **drug** dealer, with the license to **kill**, deal in drugs, and do whatever is required for his identity to remain secret, with the ultimate purpose of reporting back to his supervisors.

A con man, Irving Rosenfeld, along with his seductive partner Sydney Prosser, is **forced to work** for a wild F.B.I. Agent, Richie DiMaso, who pushes them into a world of Jersey powerbrokers and the **Mafia**

A girl in a small town forms an unlikely **bond** with a recently-paralyzed man she's **taking care** of

Enfoque "**experimental**":

```
- User name: @theashoxford
- User sentiment: Positive
- Recomendations by similar users:
        - Cast Away (2000) : ( Adventure|Drama|Romance ):  1
        - The Irishman (2019) : ( Biography|Crime|Drama|History|Thriller ):  1
        - Once Upon a Time ...in Hollywood (2019) : ( Comedy|Drama ):  1
        - Marriage Story (2019) : ( Comedy|Drama ):  1
        - The King (2019) : ( Biography|Drama|History|Romance|War ):  -1
        - The Two Popes (2019) : ( Biography|Comedy|Drama ):  1
        - A Perfect World (1993) : ( Crime|Drama|Thriller ):  -1
        - Jersey Girl (2004) : ( Comedy|Drama|Romance ):  1
        - My One and Only (2009) : ( Adventure|Biography|Comedy|Drama|Romance ):  -1
        - Macbeth (2015) : ( Drama|History|War ):  1
        - Amadeus (1984) : ( Biography|Drama|History|Music ):  -1
        - Ard al-Khof (1999) : ( Thriller ):  -1
        - Hotel Rwanda (2004) : ( Biography|Drama|History|War ):  -1
        - Eat Pray Love (2010) : ( Drama|Romance ):  -1
        - American Hustle (2013) : ( Crime|Drama ):  1
        - Birdman (2014) : ( Comedy|Drama ):  1
        - Me and Earl and the Dying Girl (2015) : ( Comedy|Drama|Romance ):  -1
        - Persepolis (2007) : ( Animation|Biography|Drama|History|War ):  1
        - Me Before You (2016) : ( Drama|Romance ):  1
        - The Dressmaker (2015) : ( Comedy|Drama ):  1
        - A Hologram for the King (2016) : ( Comedy|Drama|Romance ):  -1
        - Septembers of Shiraz (2015) : ( Thriller ):  -1
        - Lady Bird (2017) : ( Comedy|Drama ):  1
```

Recomendador final

```
def sistema_recomendador(user_id=None, max_recomendations=50, num_tweets=10, num_neighbors=7,
                         genres_sentiment=sentimiento_generos, model=loaded_model,
                         show_genres=False, show_sentiment=False):
  if user_id == None:
    user_id = input('User ID: ')

  ## Analisis de sentimiento
  user, sentiment = sentiment_from_tweets(user_id, num_tweets)
  if user == None:
    return

  ## Recomendador de pelis
  movie_list, movie_titles, movie_genres = movie_recommender(user_id, num_neighbors)

  ## Sentimiento de peliculas (por generos)
  movie_genre_sentiment = filter_genres_sentiment(movie_genres, genres_sentiment)

  ## Sentimiento de peliculas (por sinopsis)
  movie_plot_sentiment = filter_plot_sentiment(movie_list, model)


  ## Mostrar resultados:
```

```
User ID: 1
----------------------------------------------------
- User name: @Waffaboy
- User sentiment: Positive
- Recomendations by similar users:
          - The Purge: Election Year (2016)
          - The Colony (2013)
          - Lake Placid: Legacy (2018)
          - Cloverfield (2008)
          - Morgan (2016)
          - The Purge: Anarchy (2014)
- Recomendations by users and genre sentiment:
          - The Purge: Election Year (2016)
          - The Colony (2013)
          - Lake Placid: Legacy (2018)
          - Cloverfield (2008)
          - Morgan (2016)
          - The Purge: Anarchy (2014)
- Recomendations by users and plot sentiment:
          - Cloverfield (2008)
          - Morgan (2016)
----------------------------------------------------
```

```
User ID: 71
----------------------------------------------------
- User name: @best6789
- User sentiment: Negative
- Recomendations by similar users:
          - Man of Steel (2013)
          - Riddick (2013)
          - Escape from L.A. (1996)
          - Star Trek Into Darkness (2013)
          - The Purge (2013)
          - X: First Class (2011)
          - The Wolverine (2013)
          - The Conjuring (2013)
          - Texas Chainsaw 3D (2013)
          - Aftershock (2012)
          - G.I. Joe: Retaliation (2013)
          - Pacific Rim (2013)
- Recomendations by users and genre sentiment:
          - The Purge (2013)
          - The Conjuring (2013)
          - Texas Chainsaw 3D (2013)
          - Aftershock (2012)
- Recomendations by users and plot sentiment:
          - Riddick (2013)
          - Escape from L.A. (1996)
          - X: First Class (2011)
          - The Conjuring (2013)
          - Aftershock (2012)
          - G.I. Joe: Retaliation (2013)
----------------------------------------------------
```

FIN

Gracias por vuestra atención