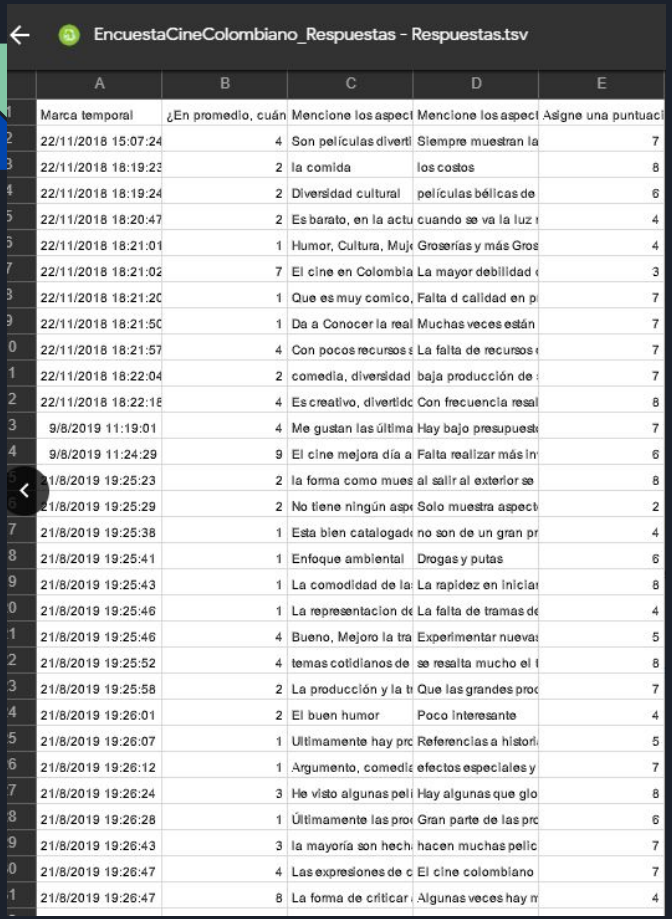
NLTK Classifier

Daniel Steven Vargas Guzman



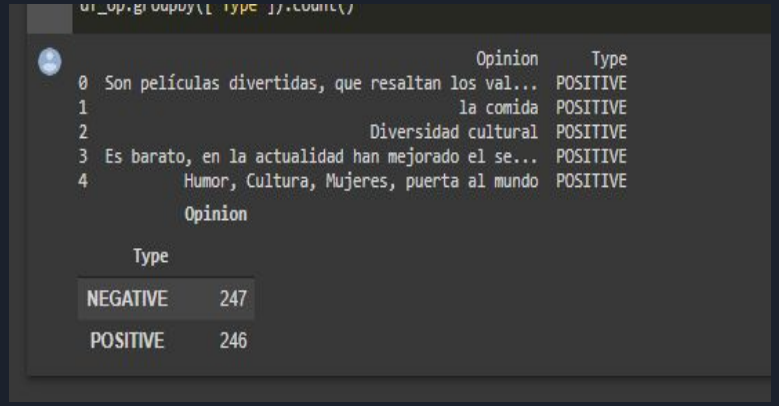
EncuestaCineColombiano_Respuestas - Respuestas.tsv

	A	B	C	D	E
1	Marca temporal	¿En promedio, cuán	Mencione los aspect	Mencione los aspect	Asigne una puntuaci
2	22/11/2018 15:07:24	4	Son películas diverti	Siempre muestran la	7
3	22/11/2018 18:19:23	2	la comida	los costos	8
4	22/11/2018 18:19:24	2	Diversidad cultural	películas bélicas de	6
5	22/11/2018 18:20:47	2	Es barato, en la actu	cuando se va la luz	4
6	22/11/2018 18:21:01	1	Humor, Cultura, Muje	Groserías y más Gros	4
7	22/11/2018 18:21:02	7	El cine en Colombia	La mayor debilidad	3
8	22/11/2018 18:21:20	1	Que es muy comico,	Falta d calidad en p	7
9	22/11/2018 18:21:50	1	Da a Conocer la real	Muchas veces están	7
10	22/11/2018 18:21:57	4	Con pocos recursos s	La falta de recursos	7
11	22/11/2018 18:22:04	2	comedia, diversidad	baja producción de	7
12	22/11/2018 18:22:18	4	Es creativo, divertido	Con frecuencia resal	8
13	9/8/2019 11:19:01	4	Me gustan las última	Hay bajo presupuest	7
14	9/8/2019 11:24:29	9	El cine mejora día a	Falta realizar más in	6
15	21/8/2019 19:25:23	2	la forma como mues	al salir al exterior se	8
16	21/8/2019 19:25:29	2	No tiene ningún aspe	Solo muestra aspect	2
17	21/8/2019 19:25:38	1	Esta bien catalogade	no son de un gran pr	4
18	21/8/2019 19:25:41	1	Enfoque ambiental	Drogas y putas	6
19	21/8/2019 19:25:43	1	La comodidad de las	La rapidez en iniciar	8
20	21/8/2019 19:25:46	1	La representacion de	La falta de tramas de	4
21	21/8/2019 19:25:46	4	Bueno, Mejoro la tra	Experimentar nuevas	5
22	21/8/2019 19:25:52	4	temas cotidianos de	se resalta mucho el t	8
23	21/8/2019 19:25:58	2	La producción y la tr	Que las grandes proc	7
24	21/8/2019 19:26:01	2	El buen humor	Poco interesante	4
25	21/8/2019 19:26:07	1	Ultimamente hay pro	Referencias a histori	5
26	21/8/2019 19:26:12	1	Argumento, comedia	efectos especiales y	7
27	21/8/2019 19:26:24	3	He visto algunas peli	Hay algunas que glo	8
28	21/8/2019 19:26:28	1	Últimamente las proc	Gran parte de las pro	6
29	21/8/2019 19:26:43	3	la mayoría son hecha	hacen muchas pelic	7
30	21/8/2019 19:26:47	4	Las expresiones de c	El cine colombiano	7
31	21/8/2019 19:26:47	8	La forma de criticar	Algunas veces hay m	4

Data information

We have a large amount of data represented as opinions about Colombian cinema, at this point, aspects such as good and bad characteristics of this topic in question remain, finally the information that is issued regarding Colombian cinema (quality)

Methodology



	Opinion	Type
0	Son películas divertidas, que resaltan los val...	POSITIVE
1	la comida	POSITIVE
2	Diversidad cultural	POSITIVE
3	Es barato, en la actualidad han mejorado el se...	POSITIVE
4	Humor, Cultura, Mujeres, puerta al mundo	POSITIVE

Type	Opinion
NEGATIVE	247
POSITIVE	246

we capture the information required in the tcv to obtain the good and bad opinions with the classification if it is a good or bad perception

1. Define a text preprocessing data

Step 2: Define a text preprocessing function

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer
from nltk.corpus import stopwords

nltk.download('stopwords')
stemmer = SnowballStemmer('spanish')
nltk.download('punkt')

stop_words = set(stopwords.words('spanish'))
stop_words = stop_words.union(set(['me', 'le', 'da', 'mi', 'su', 'ha', 'he', 'ya', 'un', 'una', 'es','del', 'las', 'los', 'en', 'que', 'y', 'la','de']))

def fast_preproc(text):
  words = word_tokenize(text.lower())
  words = [word for word in words if not word in stop_words]
  words = [stemmer.stem(word) for word in words]
  try:
    text = " ".join(str(word) for word in words)
  except Exception as e:
    print(e)
    pass
  return text

print(fast_preproc("La que me gusta"))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
gust
```

We define the function of word processing

We obtain all the "stop words" to be able to make the union with some words defined for the text processing, for this point the information is tokenized, allowing to have a perception of which are the words that should be classified with NLTK library

2. Text Preprocessing

Step 3: Text preprocessing

```
df_op['Opinion'] = df_op['Opinion'].astype(str)

df_op = df_op.assign(
    TextPreproc=lambda df: df_op.Opinion.apply(fast_preproc)
)

df_op.head()
```

	Opinion	Type	TextPreproc
0	Son películas divertidas, que resaltan los val...	POSITIVE	pelicul divert , resalt valor comport tipic co...
1	la comida	POSITIVE	com
2	Diversidad cultural	POSITIVE	divers cultural
3	Es barato, en la actualidad han mejorado el se...	POSITIVE	barat , actual mejor servici personaliz sill c...
4	Humor, Cultura, Mujeres, puerta al mundo	POSITIVE	humor , cultur , mujer , puert mund

Step 4: Split dataset

At this point we transform the information in such a way that it is recognized as a string and the data is preprocessed with the functions that are expressed in the code.

4. Split information

Step 4: Split dataset

```
#Split dataset
X = df_op['TextPreproc']
Y = df_op['Type']
```

we divide the information with the split function

Data transformation with TF_IDF

Step 5: Data transformation with TF_IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer

vec = TfidfVectorizer()

#Tokenize and build vocabulary
vec.fit(X)
print("\nVocabulary:\n")
print(vec.vocabulary_)

#Encode documents
trans_text_train = vec.transform(X)

#Print Document-Term Matrix
df = pd.DataFrame(trans_text_train.toarray(), columns=vec.get_feature_names())
df.head()
```

Vocabulary:

{'pelicul': 572, 'divert': 246, 'resalt': 671, 'valor': 783, 'comport': 166, 'tipic': 758, 'colombian': 150, 'com': 152, 'divers': 244,

	30	abarc	abord	abraz	absurd	aburr	aburridor	accesibil	accion	acerc	acert	acerv	aclam	acpect	actor	actoral	actric	actu
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
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	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 812 columns

+ Código

We carry out the vectorization of the information to convert it into a vocabulary, this with the vectorized transformation implies the immersion of the information in a dataframe

6. Split test-train datasets

Step 6: Split Test-Train Datasets

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, Y_train, Y_test = train_test_split(df, Y, test_size=0.15)
    print("Train and test shapes:\n")
    print("Train. X: " + str(X_train.shape) + " - Y:" + str(Y_train.shape))
    print("Test. X: " + str(X_test.shape) + " - Y:" + str(Y_test.shape))
```

```
Train and test shapes:

Train. X: (419, 812) - Y:(419,)
Test. X: (75, 812) - Y:(75,)
```

We carry out the training of the previously vectorized dataframe, to adjust a correct interweaving of the model for the classification of the opinions that have been transformed

7. Built Classifier

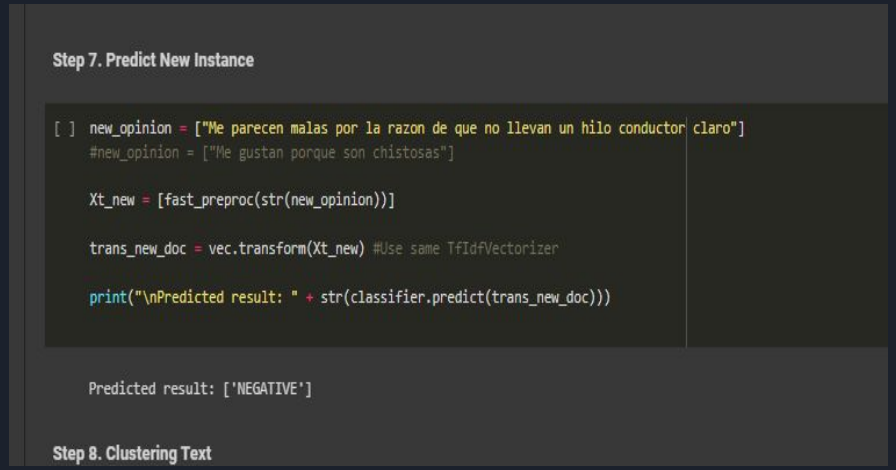
Step 7. Build a Classifier

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(solver='lbfgs')
classifier.fit(X_train, Y_train)

from sklearn.metrics import classification_report,confusion_matrix

#Predice para el conjunto de testeo.
y_pred = classifier.predict(X_test)

print("\nMatriz de confusión:\n")
print(confusion_matrix(Y_test,y_pred))

print("\nEstadisticas del clasificador:\n")
print(classification_report(Y_test,y_pred))
```

```
Matriz de confusión:

[[28 10]
 [ 2 35]]

Estadisticas del clasificador:

              precision    recall  f1-score   support

    NEGATIVE       0.93      0.74      0.82        38
    POSITIVE       0.78      0.95      0.85        37

    accuracy                           0.84        75
   macro avg       0.86      0.84      0.84        75
weighted avg       0.86      0.84      0.84        75
```

At this point, the classifier is built where all the opinions are obtained together with the sklearn.metrics library, showing their statistical values for each of the trained points and their level of precision.

8. Predict new instance



Step 7. Predict New Instance

```
[ ]  new_opinion = ["Me parecen malas por la razon de que no llevan un hilo conductor claro"]
     #new_opinion = ["Me gustan porque son chistosas"]

     Xt_new = [fast_preproc(str(new_opinion))]

     trans_new_doc = vec.transform(Xt_new) #Use same TfIdfVectorizer

     print("\nPredicted result: " + str(classifier.predict(trans_new_doc)))
```

```
Predicted result: ['NEGATIVE']
```

Step 8. Clustering Text

Previously trained the model with the pertinent libraries, we used the opinions as input, to visualize their classification according to the model, implying what the perception about Colombian cinema is, in this case the text in Spanish was used "`Me parecen malas por la razon de que no llevan un hilo conductor claro`" and the result of the perception was negative

9. Clustering

Step 8. Clustering Text

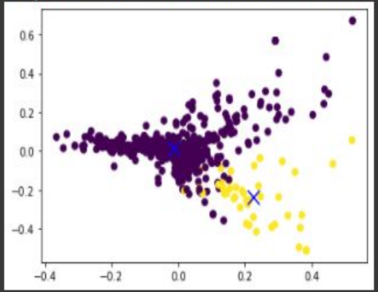
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import numpy as np

X = df_op['TextPreproc']

vec = TfidfVectorizer()
vec.fit(X)
trans_text_train = vec.transform(X)

cls = KMeans(n_clusters=2, random_state=0)
cls.fit(trans_text_train)

# Asign cluster for each sample
cls.predict(trans_text_train)

print(cls.labels_) #0: First cluster, 1: Second cluster, 2: Third cluster...
print(cls.labels_.shape)
print(sum(cls.labels_[0:246]))
print(sum(cls.labels_[247:494]))

# Use PCA in order to reduce the features to 2D
pca = PCA(n_components=2, random_state=0)
reduced_features = pca.fit_transform(trans_text_train.toarray())

# Reduce the cluster centers to 2D
reduced_cluster_centers = pca.transform(cls.cluster_centers_)

plt.scatter(reduced_features[:,0], reduced_features[:,1], c=cls.predict(trans_text_train))
plt.scatter(reduced_cluster_centers[:, 0], reduced_cluster_centers[:,1], marker='x', s=150, c='b')
```

With the training of the model and its previous validation, it is now able to interpret the text and classify it so that it can understand if it is a good or bad opinion according to the model that was built with the survey of perceptions of Colombian cinema

1. Clustering result

```
plt.scatter(reduced_cluster_centers[:, 0], reduced_cluster_centers[:,1], marker='x', s=150, c='b')
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 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 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 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 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 0 0 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 0 0 0
 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 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 0 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 0 0 1 1 0 1 0 0 0 0 0 0 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
 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 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 0 0
 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 0 1 0 0 0 1 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
(724,)
15
19
<matplotlib.collections.PathCollection at 0x7f23f77d3190>
```



Conclusion

In conclusion, with the definition of a classification model (NLTK) for the preprocessing of data (text), we can classify them according to a training model that emits a predictive response about the possible result of the opinion of perception about Colombian cinema.