Informe Tarea 3 – ISIS 4221

Universidad de los Andes

Procesamiento de Lenguaje Natural

ISIS 4221

Juan Esteban Arboleda – 201921578

Luccas Rojas – 201923052

Contenido

Punto 1

Regresión logística

Para el pipeline de regresión logística, utilizamos 2 pipelines, uno que hacía el tokenizado por medio de un TfidfVectorizer y otro por medio de un CountVectorizer. En ambos casos el tokenizador además cumple la función de normalizar el texto y eliminar las palabras de parada.

```python
lr_pipeline_tf_idf = Pipeline(
    steps=[('vectorize', TfidfVectorizer(stop_words='english',lowercase=True)),
           ('model', LogisticRegression())
    ]
)

lr_pipeline_bow = Pipeline(
    steps=[('vectorize', CountVectorizer(stop_words='english',lowercase=True)),
           ('model', LogisticRegression())
    ]
)
```

Los resultados obtenidos por cada uno de estos pipelines se puede observar en las siguientes imágenes.

Tokenizacion tf-idf regresión logistica:

	precision	recall	f1-score	support
alt.atheism	0.87	0.89	0.88	220
comp.graphics	0.79	0.84	0.82	303
comp.os.ms-windows.misc	0.78	0.84	0.81	280
comp.sys.ibm.pc.hardware	0.79	0.80	0.80	286
comp.sys.mac.hardware	0.85	0.89	0.87	275
comp.windows.x	0.89	0.84	0.86	300
misc.forsale	0.83	0.86	0.85	287
rec.autos	0.92	0.91	0.92	302
rec.motorcycles	0.97	0.97	0.97	317
rec.sport.baseball	0.96	0.96	0.96	300
rec.sport.hockey	0.97	0.97	0.97	297
sci.crypt	0.97	0.93	0.95	297
sci.electronics	0.86	0.86	0.86	318
sci.med	0.94	0.95	0.94	295
sci.space	0.90	0.95	0.92	291
soc.religion.christian	0.90	0.93	0.92	332
talk.politics.guns	0.91	0.95	0.93	252
talk.politics.mideast	0.98	0.97	0.97	294
talk.politics.misc	0.92	0.89	0.91	221
talk.religion.misc	0.93	0.60	0.73	182
accuracy			0.90	5649
macro avg	0.90	0.89	0.89	5649
weighted avg	0.90	0.90	0.89	5649

Tokenizacion tf regression logistica:

	precision	recall	f1-score	support
alt.atheism	0.86	0.86	0.86	220
comp.graphics	0.80	0.81	0.80	303
comp.os.ms-windows.misc	0.81	0.85	0.83	280
comp.sys.ibm.pc.hardware	0.78	0.75	0.76	286
comp.sys.mac.hardware	0.83	0.86	0.84	275
comp.windows.x	0.89	0.84	0.87	300
misc.forsale	0.80	0.87	0.84	287
rec.autos	0.89	0.91	0.90	302
rec.motorcycles	0.96	0.95	0.95	317
rec.sport.baseball	0.93	0.92	0.92	300
rec.sport.hockey	0.96	0.96	0.96	297
sci.crypt	0.95	0.93	0.94	297
sci.electronics	0.83	0.85	0.84	318
sci.med	0.91	0.94	0.92	295
sci.space	0.94	0.96	0.95	291
soc.religion.christian	0.91	0.93	0.92	332
talk.politics.guns	0.91	0.92	0.91	252
talk.politics.mideast	0.97	0.94	0.95	294
talk.politics.misc	0.87	0.85	0.86	221
talk.religion.misc	0.83	0.72	0.77	182
accuracy			0.88	5649
macro avg	0.88	0.88	0.88	5649
weighted avg	0.88	0.88	0.88	5649

Naïve Bayes

Para el pipeline de naive bayes, utilizamos 2 pipelines, uno que hacía el tokenizado por medio de un TfidfVectorizer y otro por medio de un CountVectorizer. En ambos casos el tokenizador además cumple la función de normalizar el texto y eliminar las palabras de parada.

```
nb_pipeline_tf_idf = Pipeline(
    steps=[('vectorize', TfidfVectorizer(stop_words='english',lowercase=True)),
           ('model', MultinomialNB())
    ]
)

nb_pipeline_bow = Pipeline(
    steps=[('vectorize', CountVectorizer(stop_words='english',lowercase=True)),
           ('model', MultinomialNB())
    ]
)
```

Los resultados obtenidos por cada uno de estos pipelines se puede observar en las siguientes imágenes.

Tokenizacion tf-idf Naive Bayes:

	precision	recall	f1-score	support
alt.atheism	0.85	0.87	0.86	220
comp.graphics	0.89	0.79	0.84	303
comp.os.ms-windows.misc	0.80	0.86	0.83	280
comp.sys.ibm.pc.hardware	0.73	0.84	0.78	286
comp.sys.mac.hardware	0.84	0.92	0.88	275
comp.windows.x	0.95	0.86	0.90	300
misc.forsale	0.93	0.77	0.84	287
rec.autos	0.91	0.92	0.91	302
rec.motorcycles	0.97	0.98	0.97	317
rec.sport.baseball	0.97	0.93	0.95	300
rec.sport.hockey	0.91	0.99	0.95	297
sci.crypt	0.86	0.97	0.91	297
sci.electronics	0.94	0.77	0.85	318
sci.med	0.98	0.94	0.96	295
sci.space	0.87	0.97	0.92	291
soc.religion.christian	0.78	0.96	0.86	332
talk.politics.guns	0.78	0.99	0.87	252
talk.politics.mideast	0.94	0.99	0.96	294
talk.politics.misc	0.98	0.77	0.86	221
talk.religion.misc	1.00	0.29	0.45	182
accuracy			0.88	5649
macro avg	0.89	0.87	0.87	5649
weighted avg	0.89	0.88	0.88	5649

Tokenizacion tf Naive Bayes:

	precision	recall	f1-score	support
alt.atheism	0.83	0.93	0.87	220
comp.graphics	0.72	0.88	0.79	303
comp.os.ms-windows.misc	0.97	0.31	0.47	280
comp.sys.ibm.pc.hardware	0.67	0.85	0.75	286
comp.sys.mac.hardware	0.78	0.91	0.84	275
comp.windows.x	0.76	0.89	0.82	300
misc.forsale	0.91	0.74	0.82	287
rec.autos	0.94	0.93	0.93	302
rec.motorcycles	0.98	0.97	0.97	317
rec.sport.baseball	0.99	0.95	0.97	300
rec.sport.hockey	0.95	0.97	0.96	297
sci.crypt	0.91	0.95	0.93	297
sci.electronics	0.93	0.82	0.87	318
sci.med	0.97	0.94	0.95	295
sci.space	0.89	0.97	0.93	291
soc.religion.christian	0.91	0.94	0.92	332
talk.politics.guns	0.86	0.96	0.91	252
talk.politics.mideast	0.93	0.99	0.96	294
talk.politics.misc	0.82	0.90	0.86	221
talk.religion.misc	0.98	0.53	0.69	182
accuracy			0.87	5649
macro avg	0.88	0.87	0.86	5649
weighted avg	0.89	0.87	0.87	5649

Cross-validation

Definicion

La validación cruzada o cross-validation sirve para diferentes cosas, lo primero es que ayuda a partir el set de entrenamiento en varias secciones normalmente denominadas pliegues, estas partes son usadas entonces para de manera alternada usar n pliegues para train y m pliegues

para validación del modelo. Esto busca sacar mejores métricas el modelo usado y disminuir el sesgo que puede traer sólo usar un train y un test para extraer las métricas de un modelo. De este modo se ejecutan múltiples veces entrenamiento y validación para tener mejores métricas que representen mejor el modelo usado. Además, este proceso puede ser usado para evaluar también cuáles son los mejores posibles hiper-parametros para un modelo dado haciendo uso de la herramienta grid-search.

Métricas

Para el modelo de regresión logística con tokenización tf-idf (mejor que tf)el resultado obtenido fue el siguiente:

```
precision_score: 0.9215582782496892
recall_score: 0.9182266793756046
f1_macro: 0.9193161533571594
f1_micro: 0.9215790405381483
```

Además de acuerdo al GridSearch realizado, el mejor hiperparámetro para c fue de 10, maximizando así la precisión del algoritmo:

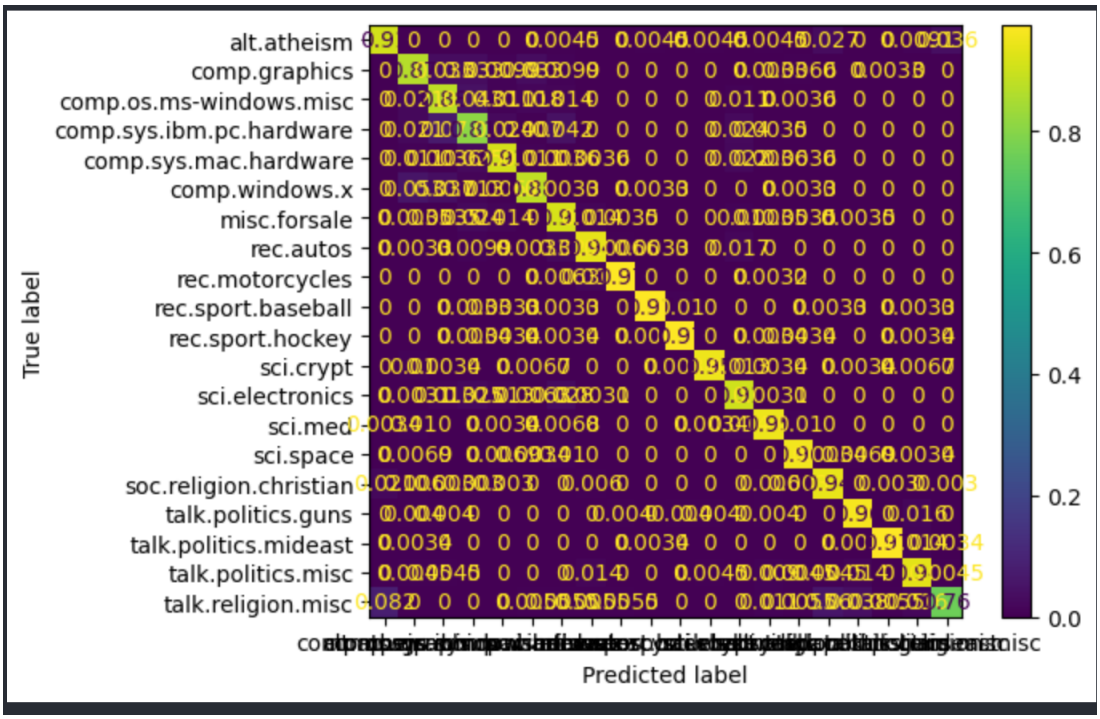
```
Score: 0.9215790405381483
Best params: {'model__C': 10}
```

Para el modelo de naive bayes con tokenización tf-idf (mejor que tf) las métricas obtenidas fueron las siguientes:

```
precision_score:0.8958206008318246
recall_score:0.8743328335705005
f1_macro:0.8731274052742443
f1_micro:0.8724181272742443
```

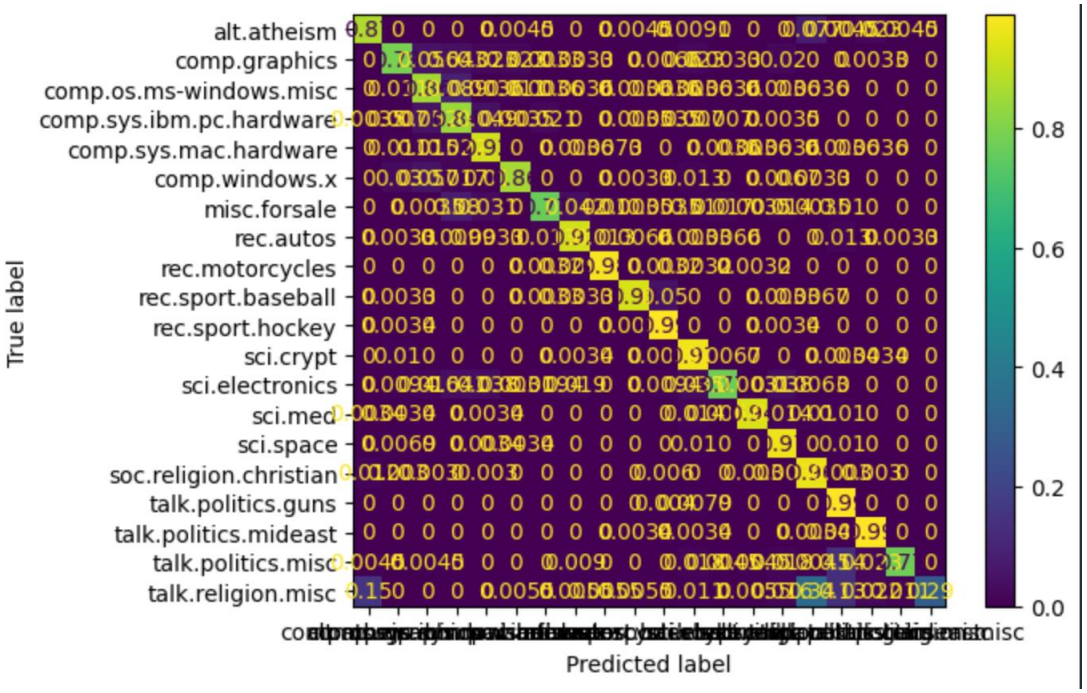
Resultados de los modelos

Resultados regression logistica




```
precision_score: 0.9215582782496892
recall_score: 0.9182266793756046
f1_macro: 0.9193161533571594
f1_micro: 0.9215790405381483
```

Resultados Naïve Bayes




```
precision_score: 0.8921136847279326
recall_score: 0.8688881772534041
f1_macro: 0.8672621347276926
f1_micro: 0.8813949371570189
```

Conclusion de resultados

A partir de los resultados obtenidos anteriormente y las matrices de confusión podemos notar como el modelo de regresión logística es superior al modelo de naive bayes, además basta aclarar que el modelo de regresión logística debe ser entrenado con un c de 10 para obtener dichos resultados. Es evidente como la regresión logística tiene mejores métricas en todos los aspectos, en precisión, recall y f1 macro y micro, lo que nos da a decir que es un modelo mas apropiado para el problema de clasificación.

Punto 2

Representación vectorial de los datos

Para este punto, fue necesario crear representaciones vectoriales de documentos. Se utilizaron 3 estrategias.

Representación TF

Para la representación TF se contó la ocurrencia de cada término por cada documento. Para los términos desconocidos se aplicó suavizado de Good-Touring.

Modelos por categorías

Para la primera parte del punto 2 se construyó un modelo de clasificación Naive Bayer y un modelo de clasificación de regresión logística para cada categoría (books, dvd, electronics, kitchen). Además, se utilizaron 3 representaciones vectoriales de los documentos: tf, tfidf y matriz de características. A continuación, se explica la manera en que dichas representaciones vectoriales fueron construidas y, posteriormente, la forma de implementación y evaluación de los modelos.

Representación TF

Para la TF se contó la ocurrencia de cada término en cada documento. Adicionalmente, se utilizó suavizado de Good-Touring para los términos desconocidos.

Representación TFIDF

Para cada documento se calculó el valor de TFIDF. Se utilizó suavizado de Good-Touring para los términos desconocidos.

Matriz de características

Utilizando el lexicón de "senticnet" se construyó una matriz con las siguientes características, por cada documento.

- Recuento de términos positivos del documento (utilizando "polarity label" del lexicón).
- Recuento de términos negativos (utilizando "polarity label" del lexicón).
- Recuento de términos por cada tipo de ánimo, del documento (utilizando "primary mood" del lexicón).

Modelo Naive Bayes

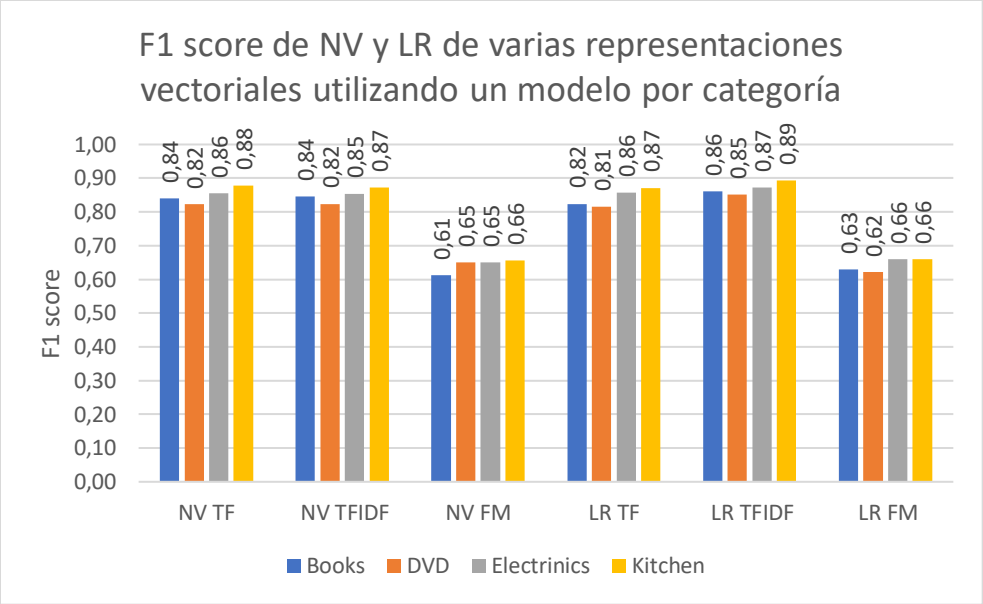
Por cada categoría, se tomaron los datos etiquetados para entrenar un modelo de Naive Bayes. Para lo anterior, se utilizó la librería "sklearn.naive_bayes.MultinomialNV". Dicha librería utiliza suavizado de Laplace.

Modelo de regresión logística

Para cada categoría se tomaron los datos etiquetados para entrenar un modelo de regresión logística. Para lo anterior, se utilizó la librería "sklearn.linear_model.LogisticRegresion". Se utilizó la penalidad por defecto de la librería (L2), para evitar overfitting.

Evaluación

Para evaluar los modelos, se utilizaron las métricas de precisión, recall, F1 score y accuracy. A continuación, se presenta el F1 score de los diferentes modelos. En los anexos de este documento puede consultar una tabla con el reporte completo de métricas de cada uno de los modelos.



NV quiere decir Naive Bayes y LR quiere decir Logistic Regression. Por su parte, TF hace referencia a la representación vectorial de frecuencias, TFIDF hace referencia a la representación vectorial TFIDF y FM hace referencia a la representación vectorial de matriz de características (Feature Matrix).

Observe que, para las tres representaciones vectoriales, y para las cuatro categorías, el desempeño de los modelos NV es bastante similar al de los modelos LR. Análogamente, las representaciones TF y TFIDF presentan resultados buenos y similares para los dos tipos de modelo y en ambas categorías. Sin embargo, la representación de matriz de características no presenta buenos resultados en ninguna categoría y bajo ningún modelo. Lo anterior, puede deberse a que se pierde mucha información del documento cuando se transforma a la representación de características. Además, muchos de los términos del corpus no existían en el lexicón, lo que también puede afectar el rendimiento de esta representación. Finalmente, aunque el rendimiento de los modelos por categoría es similar, los modelos de la categoría libros (“books”) fueron los que peor rendimiento presentaron en general. Lo anterior puede deberse a que la calificación de un libro puede ser más elaborada que la de otro producto.

Para los modelos de regresión logística, se obtuvieron las características de mayor relevancia (es decir, las características a las que el modelo les asignó un mayor peso). A continuación, se presentan las 5 características más relevantes ordenadas de mayor importancia a menor importancia.

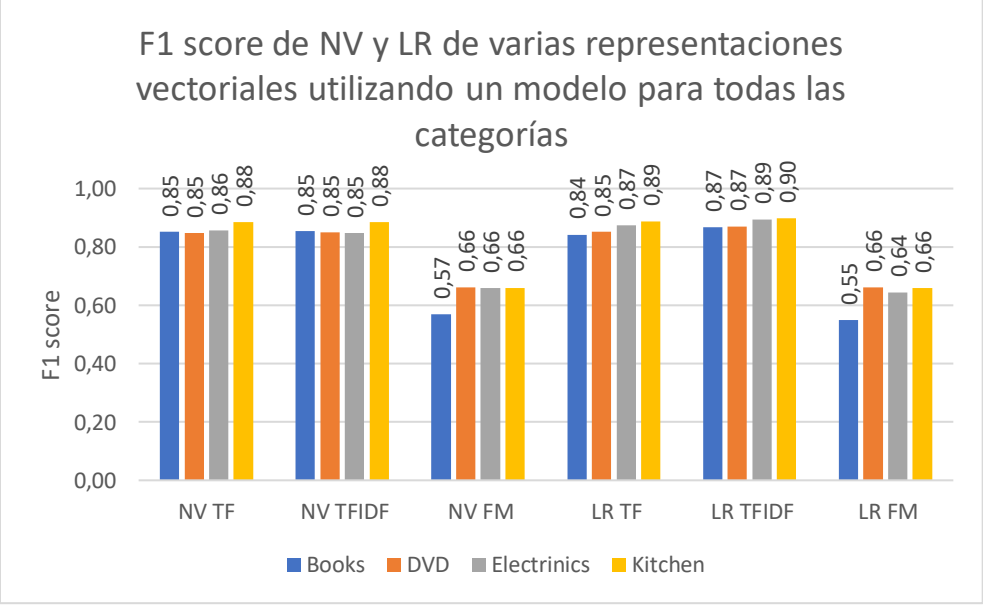
		1	2	3	4	5
Books	TF	excellent	bad	boring	disappointing	easy
	TFIDF	excellent	bad	great	waste	boring
	FM	Negative word count	#fear	#surprise	Positive word count	#interest
DVD	TF	bad	worst	boring	excellent	terrible
	TFIDF	bad	great	excellent	worst	best
	FM	#sadness	#anger	#joy	#surprise	#fear
Electronics	TF	excellent	poor	perfect	bad	great

	TFIDF	great	excellent	not	price	perfect
	FM	#disgust	#admiration	#joy	#interest	#surprise
Kitchen	TF	easy	great	perfect	excellent	best
	TFIDF	great	easy	not	love	easy_to
	FM	#disgust	Negative word count	#admiration	#anger	#fear

Para las representaciones TF y TFIDF, los features son palabras del vocabulario. Para la representación de matriz de características (FM), los features que empiezan por "#" (por ejemplo "#disgust") hacen referencia a ánimos (moods) mientras que "negative word count" y "positive word count" hacen referencia al coteo de palabras negativas y positivas respectivamente.

Modelo para todas las categorías

Para la segunda parte del punto 2, se construyó un corpus unificado con los datos etiquetados de todas las categorías. Además, se construyeron las mismas tres representaciones vectoriales mencionadas anteriormente. Con los anteriores insumos, se entrenó un modelo de clasificación Naive Bayes y un modelo de clasificación de regresión logística, por cada tipo de representación vectorial. Dichos modelos fueron construidos con las mismas librerías ya mencionadas. Posteriormente, se tomaron los datos de prueba de cada una de las categorías, y se evaluaron los modelos por cada categoría. A continuación, se presentan los resultados.



Observe que, nuevamente, el rendimiento por tipo de modelo (NV o LR) es bastante similar. Igualmente, el rendimiento de las representaciones TF y TFIDF es similar y bueno, mientras que el rendimiento de la representación de matriz de características es malo, en general. Finalmente, es importante resaltar que, el rendimiento de los modelos que utilizan el corpus unificado, aunque similar, parece ser mejor que el rendimiento de los modelos que son entrenados únicamente con el corpus de la categoría en la que son evaluados. Así, no vale la

pena construir un clasificador por cada categoría, porque el clasificador general (de todas las categorías) funciona igual de bien, o mejor que el clasificador específico de cada categoría.

Para los modelos de regresión logística, se obtuvieron las características de mayor relevancia (es decir, las características a las que el modelo les asignó un mayor peso). A continuación, se presentan las 5 características más relevantes ordenadas de mayor importancia a menor importancia.

	1	2	3	4	5
TF	excellent	disappointing	waste	boring	terrible
TFIDF	great	excellent	not	bad	waste
FM	Nositive word count	#sadness	#disgust	#joy	#fear

Para las representaciones TF y TFIDF, los features son palabras del vocabulario. Para la representación de matriz de características (FM), los features que empiezan por "#" (por ejemplo "#disgust") hacen referencia a ánimos (moods) mientras que "negative word count" y "positive word count" hacen referencia al coteo de palabras negativas y positivas respectivamente.

Observe que hay varios features importantes como "excellent", "disappointing", "great", "#disgust", entre otros, que también aparecen entre las características más importantes de los modelos por categorías.

Conclusiones

De los resultados se puede concluir que, en general, tanto los modelos de Naive Bayes como los de Regresión logística logran buenos y similares resultados en la terea de clasificación particular para las representaciones TF y TFIDF. Sin embargo, ambos tipos de modelo presentan malos resultados cuando se utiliza la representación de matriz de características. Posiblemente, una representación con más características extraídas del lexicón, o con un lexicón más completo mejore el rendimiento de dicha representación. Por otro lado, la categoría de cocina ("kitchen") fue la categoría que mejores resultados mostró, mientras que la categoría de libros ("books") fue la que peores resultados mostró. En otras palabras, es más fácil de predecir si una reseña de la categoría de cocina es positiva o negativa, en comparación con las otras categorías, mientras que es más difícil predecir si una reseña de un libro es positiva o negativa en comparación con otras categorías. Finalmente, el entrenamiento de los modelos utilizando un corpus unificado de todas las categorías mostró mejores o iguales rendimientos que el entrenamiento con el corpus específico de cada categoría.

Anexos

Reporte completo de métricas. Punto 2.

Modelo entrenado por cada categoría

	Por categoría																							
	Naive Bayes												Regresión logística											
	TF				TFIDF				FM				TF				TFIDF				FM			
Categoría	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A
Books	0,82	0,86	0,84	0,83	0,83	0,86	0,84	0,84	0,62	0,60	0,61	0,61	0,78	0,87	0,82	0,81	0,87	0,85	0,86	0,86	0,61	0,65	0,63	0,61
DVD	0,77	0,89	0,82	0,81	0,80	0,85	0,82	0,82	0,63	0,67	0,65	0,63	0,78	0,85	0,81	0,80	0,81	0,89	0,85	0,84	0,64	0,60	0,62	0,63
Electrinics	0,82	0,90	0,86	0,85	0,83	0,87	0,85	0,85	0,63	0,67	0,65	0,64	0,86	0,86	0,86	0,86	0,89	0,86	0,87	0,87	0,63	0,69	0,66	0,64
Kitchen	0,87	0,89	0,88	0,88	0,87	0,88	0,87	0,87	0,64	0,68	0,66	0,65	0,85	0,89	0,87	0,87	0,88	0,91	0,89	0,89	0,64	0,68	0,66	0,65

Modelo entrenado para todas las categorías

	Todas las categorías																							
	NV												LR											
	TF				TFIDF				FM				TF				TFIDF				FM			
Categoría	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A
Books	0,80	0,91	0,85	0,84	0,82	0,89	0,85	0,84	0,63	0,52	0,57	0,60	0,83	0,86	0,84	0,84	0,87	0,87	0,87	0,87	0,64	0,48	0,55	0,60
DVD	0,80	0,90	0,85	0,84	0,83	0,87	0,85	0,84	0,62	0,71	0,66	0,64	0,82	0,88	0,85	0,85	0,87	0,87	0,87	0,87	0,62	0,71	0,66	0,63
Electrinics	0,89	0,83	0,86	0,86	0,89	0,81	0,85	0,85	0,61	0,72	0,66	0,63	0,88	0,86	0,87	0,87	0,91	0,88	0,89	0,90	0,61	0,68	0,64	0,62
Kitchen	0,88	0,89	0,88	0,88	0,89	0,88	0,88	0,89	0,63	0,70	0,66	0,64	0,88	0,89	0,89	0,89	0,91	0,88	0,90	0,90	0,63	0,69	0,66	0,64

Nota: P hace referencia a la precisión, R al recall, F1 al F1 Score y A al accuracy. Por su parte, TF hace referencia a la representación vectorial de frecuencias, TFIDF a la representación TFIDF y FM a la representación de matriz de características.

.

Features más importantes en las regresiones logísticas. Punto 2

			1		2		3		4		5	
			Feature	Peso	Feature	Peso	Feature	Peso	Feature	Peso	Feature	Peso
Modelo por categorías	Books	TF	excellent	37,25	bad	31,59	boring	26,35	disappointing	22,96	easy	22,58
		TFIDF	excellent	8,24	bad	7,64	great	7,10	waste	6,71	boring	6,65
		FM	Negative word count	0,08	#fear	0,08	#surprise	0,07	Positive word count	0,07	#interest	0,07
	DVD	TF	bad	34,80	worst	30,26	boring	27,56	excellent	24,44	terrible	22,68
		TFIDF	bad	10,13	great	9,35	excellent	8,62	worst	8,47	best	8,13
		FM	#sadness	0,06	#anger	0,04	#joy	0,04	#surprise	0,03	#fear	0,02
	Electronics	TF	excellent	27,67	poor	22,95	perfect	21,72	bad	18,07	great	17,83
		TFIDF	great	12,69	excellent	9,28	not	8,30	price	8,15	perfect	7,68
		FM	#disgust	0,31	#admiration	0,14	#joy	0,14	#interest	0,13	#surprise	0,12
	Kitchen	TF	easy	25,92	great	20,87	perfect	20,86	excellent	20,01	best	19,62
		TFIDF	great	13,44	easy	10,95	not	10,04	love	9,41	easy_to	9,38
		FM	#disgust	0,27	Negative word count	0,07	#admiration	0,06	#anger	0,04	#fear	0,03
Modelo de todas las categorías		TF	excellent	76,02	disappointing	64,38	waste	61,13	boring	59,57	terrible	55,16
		TFIDF	great	20,34	excellent	17,56	not	16,28	bad	14,32	waste	13,79
		FM	Nositive word count	0,02	#sadness	0,02	#disgust	0,01	#joy	0,01	#fear	0,01

Nota: Para las representaciones TF y TFIDF, los features son palabras del vocabulario. Para la representación de matriz de características, los features que empiezan por "#" (por ejemplo "#disgust") hacen referencia a ánimos (moods) mientras que negative word count y positive word count hacen referencia al coteo de palabras negativas y positivas respectivamente.