Análisis Inteligente de Datos: Tarea 3

Iván González, Felipe Vásquez

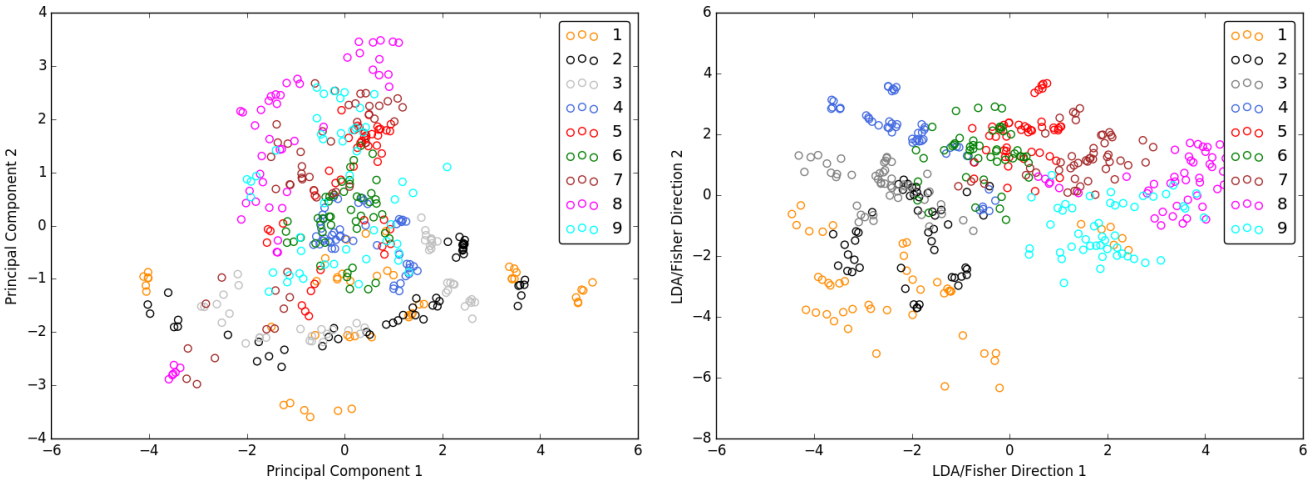
{ivan.gonzalezlo,felipe.vasquezm}@alumnos.usm.cl

14 de julio de 2016

1. Reducción de Dimensionalidad para Clasificación

a. Los dataset de entrenamiento y de pruebas están compuestos de 518 y 462 registros respectivamente.

c. Se realiza reducción de dimensionalidad utilizando PCA y LDA, de 10 a 2, lo cual puede verse en las figuras 1a y 1b respectivamente.



(a) Reducción de dimensionalidad utilizando PCA.

(b) Reducción de dimensionalidad utilizando LDA.

Figura 1: Reducción de dimensionalidad.

e. Comparando los resultados de reducción de dimensionalidad, que se encuentran en las figuras 1a (PCA) y 1b (LDA), se observa que en el caso de PCA hay una mayor sobreposición de las clases que en LDA. Lo anterior se puede explicar por el hecho de que, a diferencia de PCA, LDA toma en consideración las etiquetas de las clases, calculando las direcciones que maximizan la separación entre múltiples clases.

g. El score o exactitud de la predicción de los datos de entrenamiento y de pruebas se encuentran en la tabla 1. Para los datos de entrenamiento, QDA es el que obtiene el mejor rendimiento con un score mayor a 0.9, seguido muy de cerca por KNN. LDA se aleja más, con un score de 0.684. En tanto que para los datos de prueba, el score es similar para los tres métodos, pero con KNN siendo el mejor seguido por LDA.

En el modelo de clasificación de los k vecinos más cercanos, para un punto de consulta x_0, se buscan los k puntos de entrenamiento más cercanos en distancia a un punto x_0 y luego se clasifica usando mayoría de votos entre los k vecinos. En la figura 2, se observa la variación del score, clasificando los datos de entrenamiento y pruebas, a medida que se aumenta el número de vecinos k.

Modelo	Score datos de entrenamiento	Score datos de prueba
LDA	0.684	0.452
QDA	0.989	0.416
KNN (k=10)	0.932	0.491

Cuadro 1: Score de los modelos de clasificación utilizados.

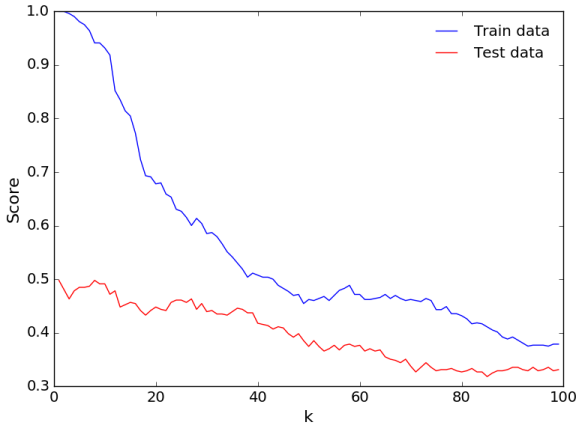
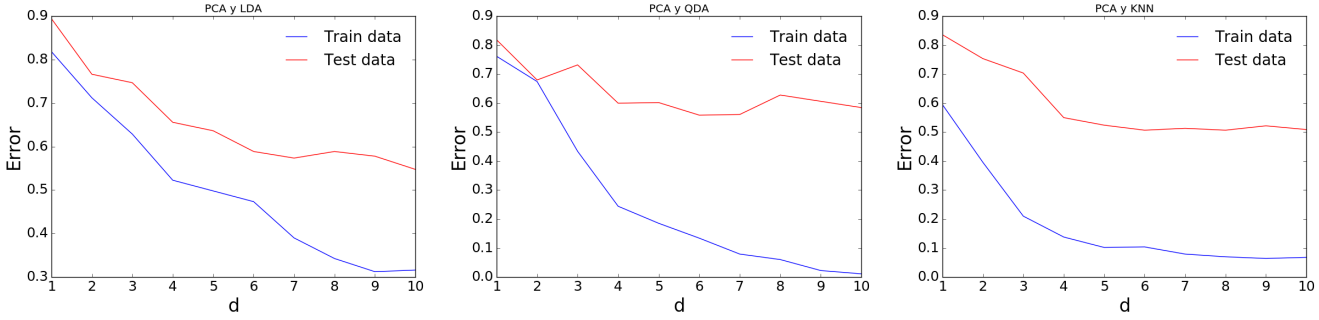


Figura 2: Score del modelo del *vecino más cercano* en función del parámetro k

h. En la figura 3, se observa que para los tres modelos de clasificación evaluados (LDA, QDA y KNN), el error de entrenamiento es monótonamente decreciente en la medida en que se consideran más componentes principales de los datos, luego de utilizar como método de reducción de dimensionalidad a PCA. En consecuencia, el error mínimo de entrenamiento se alcanza con 10 dimensiones. Comparando los dos tipos de errores (entrenamiento y pruebas), el primero es menor que el segundo en los tres casos, lo cual es natural puesto que se están clasificando los mismos datos que se utilizaron para entrenar el modelo. Respecto del error de pruebas en específico, en la tabla 2 pueden verse los errores mínimos para cada uno de los modelos, con la cantidad de dimensiones asociadas. El modelo con PCA-KNN es el que muestra los mejores resultados, entregando el error más pequeño y con la menor cantidad de dimensiones (modelo más simple).



(a) PCA y LDA. **(b)** PCA y QDA. **(c)** PCA y KNN.

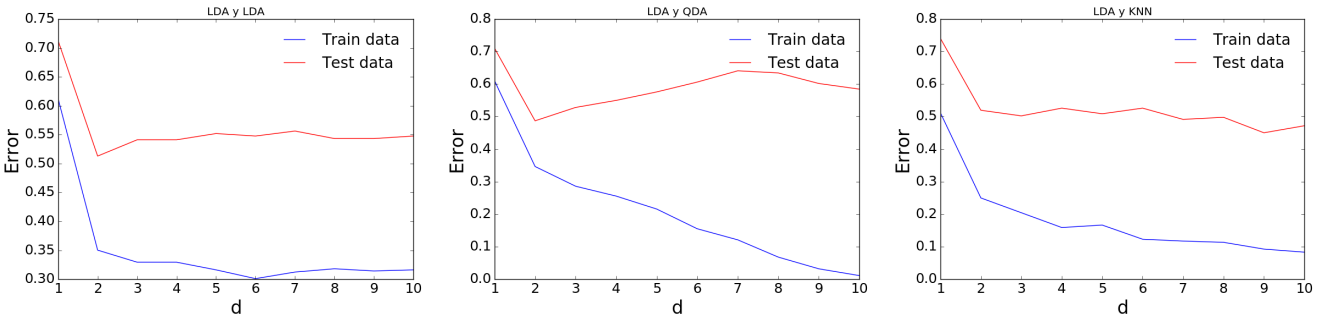
Figura 3: Variación del error ($1 - accuracy$) al cambiar la dimensionalidad con PCA.

Modelo	**Min error test**	**Dimensiones**
LDA	0,548	10
QDA	0,588	6
KNN (k=10)	0,506	6

Cuadro 2: Mínimo error de pruebas alcanzado en cada uno de los modelos, usando PCA para reducir dimensionalidad.

i. En la figura 4, se puede ver un resultado similar al observado en el ítem anterior en relación al error de entrenamiento (más dimensiones, error más pequeño). En tanto que para el error de pruebas (tabla 3), el valor más pequeño se obtuvo clasificando con KNN. Sin embargo, realizando la clasificación con QDA se obtiene un error de apenas un 8,2 % superior al que se obtuvo con KNN, pero con una cantidad mucho menor de dimensiones, 2 versus 9, por lo que la combinación LDA-QDA es la más atractiva analizando en términos conjuntos su performance y complejidad (más simple), para construir un modelo de clasificación. Finalmente,

comparando con el item anterior, los errores de prueba son menores cuando se reduce dimensionalidad con LDA que con PCA, corroborando de alguna forma lo obtenido de forma cualitativa en el item c.



(a) LDA y LDA. **(b)** LDA y QDA. **(c)** LDA y KNN.

Figura 4: Variación del error ($1 - accuracy$) al cambiar la dimensionalidad con LDA.

Modelo	Min error test	Dimensiones
LDA	0,513	2
QDA	0,487	2
KNN (k=10)	0,450	9

Cuadro 3: Mínimo error de pruebas alcanzado en cada uno de los modelos, usando LDA para reducir dimensionalidad.

2. Análisis de Opiniones sobre Películas

a. Tanto el conjunto de entrenamiento como el conjunto de prueba tienen 3554 registros para cada clase. Las clases son *Sentiment* y *Text*.

b. El *stemming* es un proceso heurístico que corta la derivación de las palabras para encontrar la raíz[1]. Por ejemplo autómata, automático, automatizado se reducen a autómata. Y en relación a los ejemplos entregados en el enunciado de la tarea, si se consideran los textos "I love eating cake" y "I loved eating the cake", con *stemming* se consigue "love eat cake" para ambos casos, mientras que sin *stemming* se obtiene "love eating cake" y "loved eating cake". En consecuencia, el vocabulario de palabras considerando ambos textos, es más pequeño si se aplica *stemming*.

c. En el caso de la *lematización*, es bastante similar al *stemming*, en el sentido que reducen las formas inflecionales de las palabras a una base común o raíz[2]. Pero en el caso de la lematización y a partir de la implementación de `nltk`[3], se hace un chequeo de la forma reducida en el corpus de WordNet. Si no está en este último, se regresa la palabra original. En consecuencia, la *lematización* es un proceso más complejo que el *stemming*. Finalmente, considerando los textos de prueba del enunciado de la tarea, por ejemplo para "I love eating cake" se obtiene "love eating cake", donde "eating" no se reduce a "eat".

d. Para ambos datasets, de entrenamiento y de pruebas, se obtuvo el top 10 de palabras más frecuentes en el vocabulario. En ambos casos se obtienen principalmente palabras relacionadas al mundo de las películas. Estas son:

[1] `http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html`
[2] Ver nota al pie 1
[3] `http://www.nltk.org/api/nltk.stem.html#nltk.stem.wordnet.WordNetLemmatizer`

Entrenamiento		Prueba	
Frecuencia	Palabra	Frecuencia	Palabra
556	film	558	film
481	movie	540	movie
246	one	250	one
245	like	238	ha
224	ha	230	like
183	make	197	story
176	story	175	character
163	character	165	time
145	comedy	161	make
143	time	134	comedy

Cuadro 4: Palabras más frecuentes en el conjunto de entrenamiento y de prueba.

e. Las métricas que calcula la función *classification_ report* son la siguientes Precisión, Recall y F1-score para cada clase

- Precisión: Cantidad de resultados positivos correctos divididos por la cantidad total de resultados positivos
- Recall: Cantidad de resultados positivos correctos dividido por el número de resultados positivos que se debería obtener.
- F1-score: Es una medida de la exactitud de una prueba, que se calcula como sigue

$$F_1 = 2 \cdot \frac{Precisión \cdot Recall}{Precisión + Recall} \tag{1}$$

f. Se obtienen las métricas relacionadas al aplicar un clasificador *Bayesiano Ingenuo Binario*, para los diferentes casos solicitados.

Métricas	Lematización	L-sin stopwords	Stemming
Training Accuracy	0.959	0.955	0.943
Test Accuracy	0.739	0.749	0.748
Precision	0.74	0.75	0.75
Recall	0.74	0.75	0.75
F1-score	0.74	0.75	0.75

Cuadro 5: Métricas obtenidas al aplicar un clasificador *Bayesiano Ingenuo Binario*, aplicando diferentes técnicas de pre-procesamiento en texto.

En términos generales, se obtienen buenos resultados utilizando lematización y *stemming* como se observa en el cuadro 5, siendo este último un poco mejor. Al trabajar sin *stopwords* el procesamiento se vuelve más lento al aumentar el vocabulario pero el resultado sigue siendo bueno. No obstante, las palabras más frecuentes son artículos, pronombres y preposiciones en su mayoría perdiendo poder de análisis en los datos.

g. Se obtienen las métricas relacionadas al aplicar un clasificador *Bayesiano Ingenuo Multinomial*, para los diferentes casos solicitados.

Métricas	Lematización	L-sin stopwords	Stemming
Training Accuracy	0.959	0.955	0.942
Test Accuracy	0.741	0.748	0.750
Precision	0.74	0.75	0.75
Recall	0.74	0.75	0.75
F1-score	0.74	0.75	0.75

Cuadro 6: Métricas obtenidas al aplicar un clasificador Bayesiano Ingenuo Multinomial, aplicando diferentes técnicas de pre-procesamiento en texto.

De acuerdo al cuadro 6 se obtienen buenos resultados utilizando lematización y *stemming*, siendo este último el que logra un mejor resultado. Al trabajar sin *stopwords* el procesamiento se vuelve más lento, debido al aumento en el vocabulario pero el resultado sigue siendo bueno. No obstante, las palabras más frecuentes son artículos, pronombres y preposiciones en su mayoría perdiendo poder de análisis en los datos.

h. La regularización permite controlar el efecto de los estimadores en la predicción, es decir, el aporte que tienen estos. Para el caso del algoritmo de Regresión logística (ocurre lo mismo en SVM lineal) utiliza como parámetro el valor inverso del parámetro de regularización, por tanto tiene un efecto contrario al habitual. Si la regularización toma un valor bajo (cercano a cero) significa que el aporte de los estimadores en la predicción será alto (la penalización es baja) y si el parámetro toma un valor alto ocurre lo contrario (penalización alta). Entonces, con la regularización adecuada se busca reducir el efecto de *overfitting* en el modelo predictivo, al reducir o disminuir la participación de los estimadores en el modelo.

   Se obtienen las métricas relacionadas al aplicar un modelo de *Regresión Logística Regularizado*, penalizando con normal l_2 y parámetro de regularización $C = 0{,}1$, para los diferentes casos solicitados.

Métricas	Lematización	L-sin stopwords	Stemming
Training Accuracy	0.892	0.879	0.880
Test Accuracy	0.719	0.719	0.731
Precision	0.72	0.72	0.73
Recall	0.72	0.72	0.73
F1-score	0.72	0.72	0.73

Cuadro 7: Métricas obtenidas al aplicar un modelo de *Regresión Logística Regularizado*, aplicando diferentes técnicas de pre-procesamiento en texto.

De acuerdo al cuadro 7, nuevamente se obtienen buenos resultados utilizando lematización y *stemming*, siendo este último el que logra un resultado mejor. Al eliminar la técnica de *stopwords* el procesamiento se vuelve más lento, debido al aumento en el vocabulario pero, el resultado sigue siendo bueno. No obstante, las palabras más frecuentes son en su mayoría artículos, pronombres y preposiciones perdiendo poder de análisis en los datos.

i. Se obtienen las métricas relacionadas al aplicar *SVM lineal*, con un parámetro de regularización de $C = 0{,}1$, para los diferentes casos solicitados.

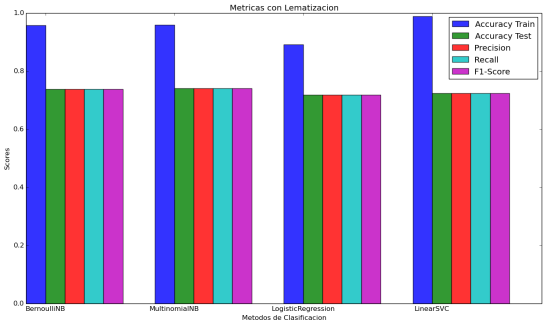
Métricas	Lematización	L-sin stopwords	Stemming
Training Accuracy	0.989	0.988	0.982
Test Accuracy	0.724	0.738	0.731
Precision	0.72	0.74	0.73
Recall	0.72	0.74	0.73
F1-score	0.72	0.74	0.73

Cuadro 8: Métricas obtenidas al aplicar *SVM lineal*, aplicando diferentes técnicas de pre-procesamiento en texto.

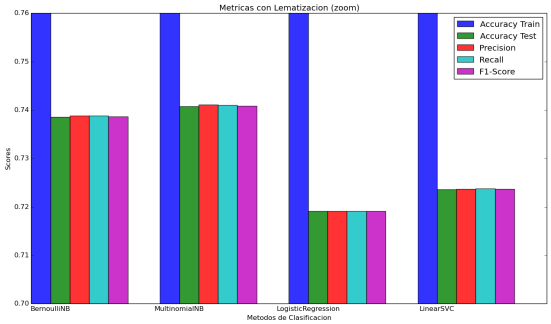
De acuerdo al cuadro 8, nuevamente se obtienen buenos resultados utilizando lematización y *stemming*, siendo este último el que logra un resultado mejor. Al eliminar la técnica de *stopwords* el procesamiento se vuelve más lento, debido al aumento en el vocabulario pero, el resultado mejora un poco debido a este aumento de información. No obstante, las palabras más frecuentes son en su mayoría artículos, pronombres y preposiciones perdiendo poder de análisis en los datos.

j. Se comparan los diferentes métodos de clasificación vistos de acuerdo a las métricas: accuracy, precisión, recall y f1-score. Utilizando las técnicas de lematización y *stemming*.

   En las figuras 5a y 6a se puede observar que los cuatro métodos tienen un comportamiento similar obteniendo buenos resultados (accuracy, precisión, recall y f1-score > 70). Para un análisis más preciso se realiza un zoom a estos gráficos, esto se ve reflejado en las figuras 5b y 6b. Aquí podemos observar que el método que obtiene mejores resultados es el clasificador *Bayesiano Ingenuo Multinomial*. No obstante, para el modelo de Regresión logística y SVM lineal no se realizo un estudio exhaustivo del parámetro de regulación ha utilizar, por lo que con un regularizador más adecuado podrían mejorar sus resultado, aun cuando estos ya son buenos.

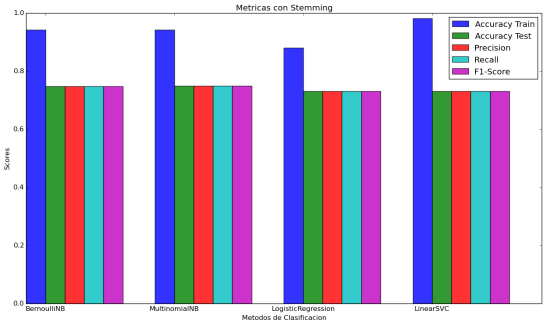


(a) Comparación de métodos utilizando lematización.

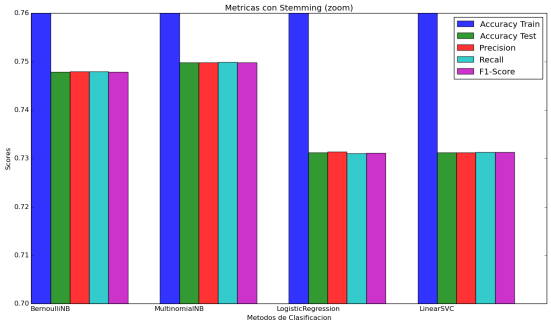


(b) Zoom de la comparación.

Figura 5: Comparación de métodos bajo las diferentes métricas evaluadas.



(a) Comparación de los métodos utilizando stemming.



(b) Zoom de la comparación.

Figura 6: Comparación de métodos bajo las diferentes métricas evaluadas.