Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

Tarea 5

Marcelo Alberto Sanchez Zaragoza

10 de mayo de 2021

1. Problema 1

Para datos de clasificación binaria $\{(x_i, y_i)\}_{i=1}^n$, considera la siguiente función de costo:

$$L = \sum_i (\theta(y_i) - \beta^t x_i - \beta_0)^2$$

Definimos n_+, n_- el número de observaciones con $y_i = 1$ y $y_i = -1$, respectivamente c_+, c_- el centroide de las observaciones con $y_i = 1$ y $y_i = -1$ y c el centroide de todos los datos.
Como en clase, construimos las matrices:

$$S_B = (c_+ - c_-)(c_+ - c_-)^t$$
$$S_W = \sum_{i:y_i=1} (x_i - c_+)(x_i - c_+)^t + \sum_{i:y_i=-1} (x_i - c_-)(x_i - c_-)^t$$

Inciso a

Verifica que

$$S_W = \sum_{i:y_i=1}^{n_+} x_i x_i^t + \sum_{i:y_i=-1}^{n_-} x_i x_i^t - n_+ c_+ c_+^t - n_- c_- c_-^t$$

Tenemos:

$$S_W = \sum_{i:y_i=1}^{n_+} (x_i - c_+)(x_i - c_+)^t + \sum_{i:y_i=-1}^{n_-} (x_i - c_-)(x_i - c_-)^t =$$
$$\sum_{i:y_i=1}^{n_+} (x_i x_i^t - 2c_+ x_i^t + c_+ c_+^t) + \sum_{i:y_i=-1}^{n_-} (x_i x_i^t - 2c_- x_i^t + c_- c_-^t)$$

Antes de continuar, sabemos:

$$c_+ = \frac{1}{n_+} \sum_{i:y=1}^{n_+} x_i; \quad c_- = \frac{1}{n_-} \sum_{i:y=-1}^{n_-} x_i$$

Regresamos podemos sustituir:

$$S_W = \sum_{i:y_i=1}^{n_+} (x_i x_i^t - 2c_+ x_i^t + c_+ c_+^t) + \sum_{i:y_i=-1}^{n_-} (x_i x_i^t - 2c_- x_i^t + c_- c_-^t)$$
$$= \sum_{i:y_i=1}^{n_+} x_i x_i^t - 2c_+(n_+ c_+^t) + n_+ c_+ c_+^t + \sum_{i:y_i=-1}^{n_-} x_i x_i^t - 2c_-(n_- c_-^t) + n_- c_- c_-^t$$
$$= \sum_{i:y_i=1}^{n_+} x_i x_i^t + \sum_{i:y_i=-1}^{n_-} x_i x_i^t - n_+ c_+ c_+^t - n_- c_- c_-^t$$

Finalmente tenemos:

$$S_W = \sum_{i:y_i=1}^{n_+} x_i x_i^t + \sum_{i:y_i=-1}^{n_-} x_i x_i^t - n_+ c_+ c_+^t - n_- c_- c_-^t$$

Inciso b
Verifica que el vector $S_B\beta$, es múltiplo del vector $(c_+ - c_-)$.

Sabemos que β contiene los coeficientes β_i de $i = 1, ..., n$ de la siguiente forma $\beta = [\beta_1, \beta_2, ..., \beta_n]^t$. También tenemos que $S_B = (c_+ - c_-)(c_+ - c_-)^t$ por lo que vamos a sustituir en $S_B\beta$, así tenemos:

$$S_W\beta = (c_+ - c_-)(c_+ - c_-)^t\beta$$

Pero podemos tomar $(c_+ - c_-)^t\beta$ y tomarla como una constante K y volver a sustuir.

$$\begin{aligned} S_W\beta = (c_+ - c_-)(c_+ - c_-)^t\beta &= (c_+ - c_-)[(c_+ - c_-)^t\beta] \\ &= (c_+ - c_-)K \end{aligned}$$

Por lo que observamos que este producto es multiplo de $(c_+ - c_-)$.

Inciso c
Si definimos $\theta(1) = n/n_+$ y $\theta(-1) = -n/n_-$, verifica que el mínimo de $L = \sum_i(\theta(y_i) - \beta^t x_i - \beta_0)^2$:

$$\beta_0 = -\beta^t c$$
$$(S_W + \frac{n_+ n_-}{n} S_B)\beta = n(c_+ - c_-)$$

Para poder trabajar el problema vamos a partir de lo siguiente: $X^tX\beta = X^tY$.
Donde:

$$X^t = \begin{pmatrix} 1 & ... & 1 & ... & 1 \\ x_{11} & ... & x_{1p} & ... & x_{1n} \\ x_{21} & ... & x_{2p} & ... & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & ... & x_{np} & ... & x_{nn} \end{pmatrix} = \begin{pmatrix} 1 & ... & 1 & ... & 1 \\ x_1 & ... & x_{n_+} & ... & x_{n_-} \end{pmatrix}$$

$$Y^t = \begin{pmatrix} n/n_+ & n/n_+ & ... & n/n_+ & n/n_- & n/n_- & ... & n/n_- \end{pmatrix}$$

$$\beta = \begin{pmatrix} \beta_0 & \beta \end{pmatrix}$$

Ya que hemos aclarado nuestras matrices vamos a comenzar con la sustitución:

$$X^tX = \begin{pmatrix} 1 & ... & 1 & ... & 1 \\ x_1 & ... & x_{n_+} & ... & x_{n_-} \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_{n_-} \end{pmatrix} = ...$$

$$= \begin{pmatrix} n & \sum_{i=1}^{n} x_i^t \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i x_i^t \end{pmatrix}$$

Pero sabemos que $\sum_{i=1}^{n} x_i = n_+c_+ + n_-c_-$, también podemos encontrar el valor de $\sum_{i=1}^{n} x_i x_i^t$ ya que en el inciso a) encontramos $S_W = \sum_{i:y_i=1}^{n_+} x_i x_i^t + \sum_{i:y_i=-1}^{n_-} x_i x_i^t - n_+c_+c_+^t - n_-c_-c_-^t$, entonces solo debemos sustituir en la matriz:

$$X^tX = \begin{pmatrix} n & n_+c_+^t + n_-c_-^t \\ n_+c_+ + n_-c_- & S_W + n_+c_+c_+^t + n_-c_-c_-^t \end{pmatrix}$$

Ahora vamos a realizar el producto de X^tY:

$$X^tY = \begin{pmatrix} 1 & ... & 1 & ... & 1 \\ x_1 & ... & x_{n_+} & ... & x_{n_-} \end{pmatrix} \begin{pmatrix} n/n_+ \\ n/n_+ \\ \vdots \\ ... \\ n/n_- \end{pmatrix} = \begin{pmatrix} n_+(n/n_+) - n_-(n/n_-) \\ \left(\sum_{i=1}^{n_+} x_i\right)(n/n_+) + \left(\sum_{i=-1}^{n_-}\right)(-n/n_-) \end{pmatrix} =$$

$$= \begin{pmatrix} 0 \\ n(c_+ - c_-) \end{pmatrix}$$

De la siguiente expresión vamos a despejar β_0 y β:

$$\begin{pmatrix} n & n_+c_+^t + n_-c_-^t \\ n_+c_+ + n_-c_- & S_W + n_+c_+c_+^t + n_-c_-c_-^t \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ n(c_+ - c_-) \end{pmatrix}$$

Así tenemos para β_0:

$$n\beta_0 + \beta(n_+c_+^t + n_-c_-^t) = 0$$

$$\beta_0 = -\frac{n_+c_+^t + n_-c_-^t}{n}\beta$$

Sabemos que $c = \frac{n_+c_+ + n_-c_-}{n}$, por lo que solo debemos transponer la expresión anterior y sustituir, así vamos a tener $\beta_0 = -c^t\beta$ pero podemos trasponer nuevamente y finalmente tendremos: $\beta_0 = -\beta^t c$.

Regresando al segundo despeje tenemos lo siguiente:

$$(n_+c_+ + n_-c_-)\beta_0 + (S_W + n_+c_+c_+^t + n_-c_-c_-^t)\beta = n(c_+ - c_-)$$

Pero ya sabemos el valor de β_0 por lo que vamos a sustituir el valor.

$$(n_+c_+ + n_-c_-)(-c^t\beta) + (S_W + n_+c_+c_+^t + n_-c_-c_-^t)\beta = n(c_+ - c_-)$$
$$(n_+c_+ + n_-c_-)(-\frac{(n_+c_+^t + n_-c_-^t)}{n}\beta) + (S_W + n_+c_+c_+^t + n_-c_-c_-^t)\beta = n(c_+ - c_-)$$
$$(-\frac{(n_+^2c_+c_+^t + 2n_+n_-c_-c_+ + n_-^2c_-c_-^t)}{n} + S_W + n_+c_+c_+^t + n_-c_-c_-^t)\beta = n(c_+ - c_-)$$
$$(S_W - \frac{(n_+^2c_+c_+^t)}{n} + n_+c_+c_+^t - \frac{(2n_+n_-c_-c_+^t)}{n} - \frac{(n_-^2c_-c_-^t)}{n} + n_-c_-c_-^t)\beta = n(c_+ - c_-)$$
$$(S_W + \frac{n_+}{n}(-n_+ + n)c_+c_+^t - \frac{(2n_+n_-)}{n}c_-c_+ + \frac{n_-}{n}(-n_- + n)c_-c_-^t)\beta = n(c_+ - c_-)$$
$$(S_W + \frac{(n_+n_-)}{n}(c_+c_+^t - 2c_-c_+^t + c_-c_-^t))\beta = n(c_+ - c_-)$$

Recordando que en clases construimos las matrices $S_B = (c_+ - c_-)(c_+ - c_-)^t$ donde al desarrollar tenemos que $S_B = c_+c_+^t - 2c_-c_+^t + c_-c_-^t$ por lo que finalmente tenemos: $(S_W + \frac{(n_+n_-)}{n}S_B)\beta = n(c_+ - c_-)$.

Inciso d

Usando el resultado del inciso b), argumaneta que (2) implica que en el mínimo:

$$\beta \approx S_W^{-1}(c_+ - c_-)$$

es decir la solución coincide con la de Fisher Discriminant Analysis(FDA).

Podemos empezar con: $(S_W + \frac{n_+ n_-}{n} S_B)\beta = n(c_+ c_-)$ y empezamos a desarrollar:

$$(S_W + \frac{n_+ n_-}{n} S_B)\beta = n(c_+ c_-)$$
$$(S_W + \frac{n_+ n_-(c_+ - c_-)(c_+ - c_-)^t}{n})\beta = n(c_+ c_-)$$

Vimos en el inciso b) que $(c_+ - c_-)^t\beta$ lo tomamos como una constante y al sustituir:

$$S_W\beta + n_+ n_-(c_+ - c_-)\frac{K}{n} = n(c_+ c_-)$$
$$S_W\beta = n(c_+ c_-) - n_+ n_-(c_+ - c_-)\frac{K}{n}$$
$$S_W\beta = (c_+ + c_-)(n - n_+ n_-\frac{K}{n})$$
$$\beta = S_W^{-1}(c_+ + c_-)(n - n_+ n_-\frac{K}{n})$$

podemos tomar $(c_+ + c_-)(n - n_+ n_- \frac{K}{n})$ como una constante y finalmente podemos escribir:

$$\beta \approx S_W^{-1}(c_+ - c_-)$$

Inciso e

El inciso nos menciona que por lo realizado en los anteriores incisos podemos implementar FDA usando algún algoritmo de mínimos cuadrados, en este caso nos apoyamos del módulo *sklearn.linear_model*.
Para ilustrar mejor la implementación se generan dos grupos de datos en dos dimensiones, cada conjunto de datos tiene 100 observaciones, se generaron por medio de una función que vienen implementada en el lenguaje de programación(PYTHON), si se desea replicar los resultados se anexaran el código que se empleo.

Una vez que hemos aclarado como generamos nuestras muestras, en la figura 1.1 tenemos los datos en dos dimensiones.

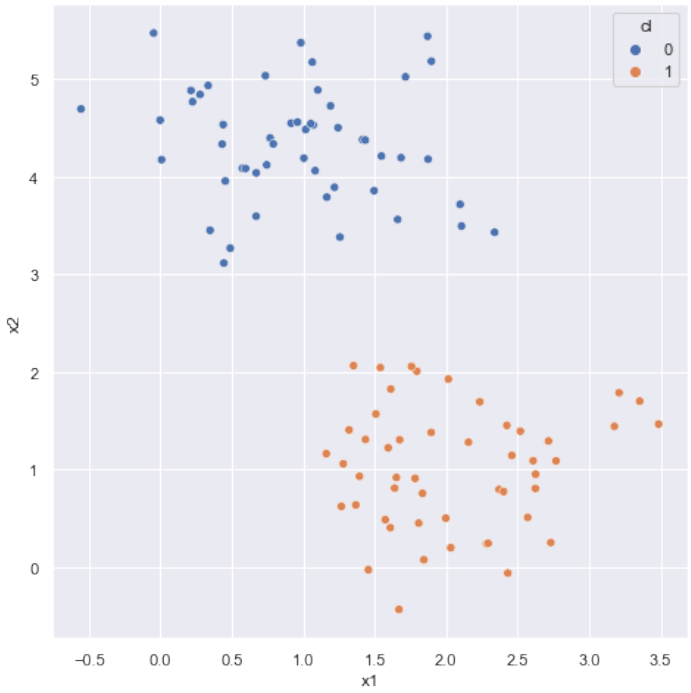

Figura 1.1: Conjuntos de datos en dos dimensiones - Ejemplo1

Los datos que estamos ocupando en la figura 1.1 no se encuentran muy alejados ya que como mencionamos al principio la intención es observar que tan buena es la implementación que estamos realizando.

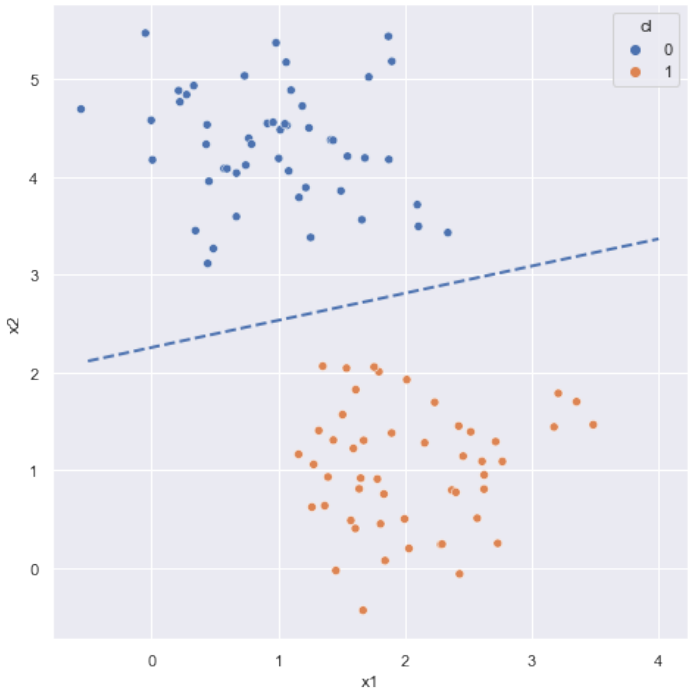

Figura 1.2: Resultado de implementar FDA por medio de mínimos cuadrados

En la figura 1.2 obsevamos que la implementación funciona muy bien por lo que ahora probaremos para otros datos.
Con el siguiente conjunto de datos los pedimos un poco más alejados por lo que en la figura 1.3 nuestros grupos se ven más lejanos.

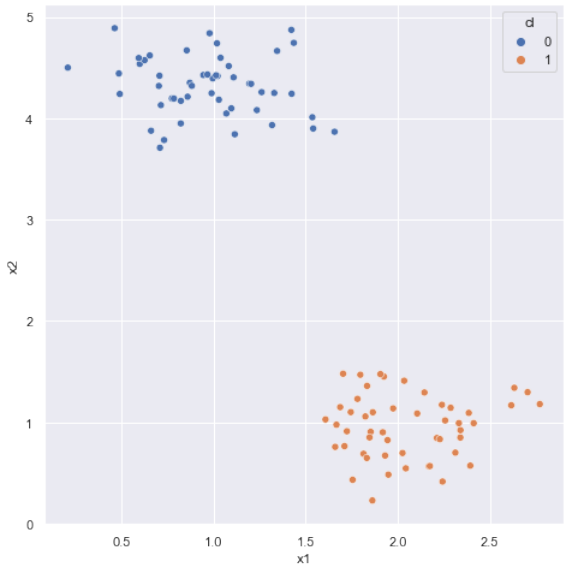

Figura 1.3: Conjuntos de datos en dos dimensiones - Ejemplo2

Al aplicar la implementación observamos que el resultado de la figura 1.4 fue muy bueno por lo que para el segundo conjunto de muestras sirvio solo para ilustrar mejor la implementación.

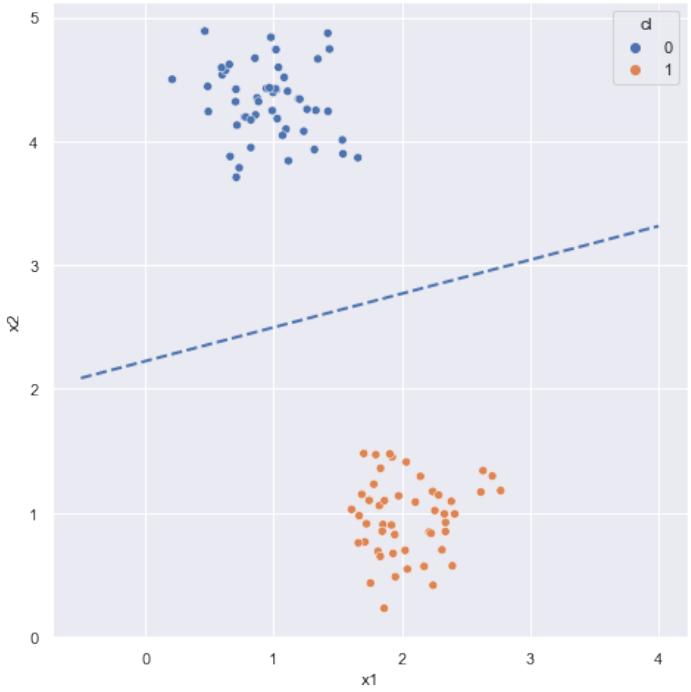

Figura 1.4: Resultado de implementar FDA por medio de mínimos cuadrados

Inciso f

Para el caso donde nuestros datos contengan datos atípicos, el plano separador cambio mucho debido a los datos atípicos que agregamos. En la figura 1.5 se muestran los datos originales y los datos atípicos que agregamos de manera manual.

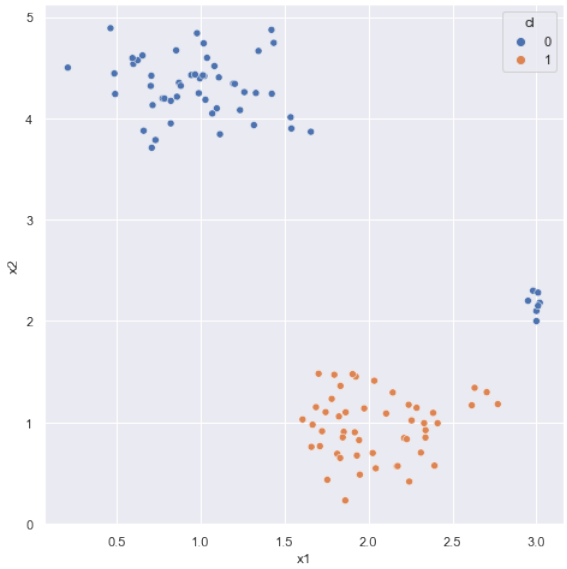

Figura 1.5: Conjuntos de datos en dos dimensiones - Datos atípicos

Al aplicar la implementación observamos que nuestro plano separador cambia bastante por lo que tenemos que hacer un ajuste a la implementacia y eso se arregla asignando pesos a los distintos datos, es decir, que tan importante son los datos para la implementación.

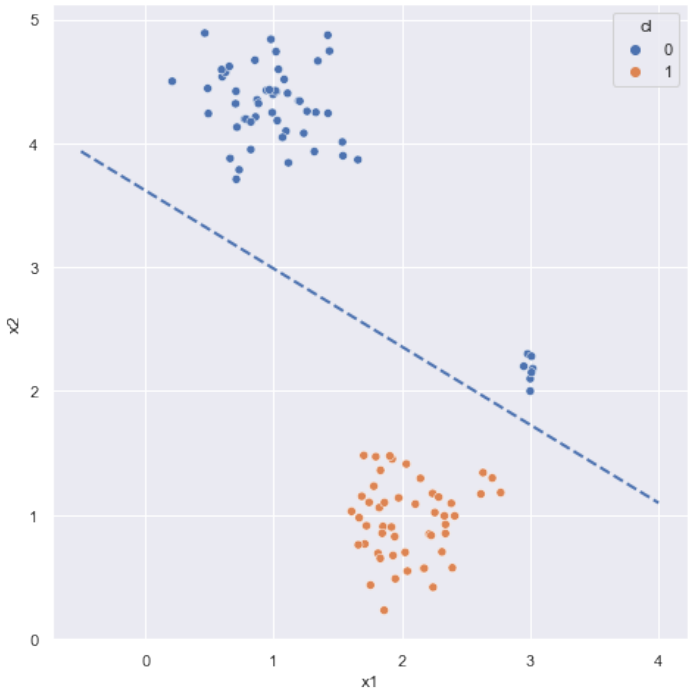

Figura 1.6: Resultado de implementar FDA por medio de mínimos cuadrados

Los pesos que vamos a asignar tienen relación con la distancia que existe entre el centroide y los datos, esto nos ayuda bastante ya que a medida que la distancia entre el centroide y los datos aumente nos da un buen indicio de que los datos posiblemente sean atípicos y no deberiamos tomarlos en cuenta.
En otras palabras, estamos escogiendo un cierto peso tal que mientras más alejados esten los datos del centroide se les da menos importancia, esto para omitir aquellas representaciones atipícas que nos puedan surgir.

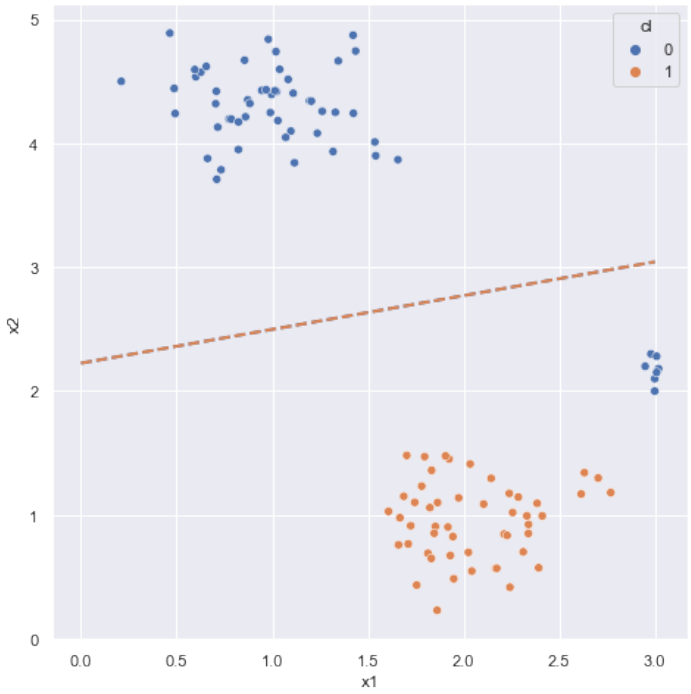

Figura 1.7: Resultado de implementar FDA con pesos

En la figura 1.7 observamos que no contempla los datos atipícos que ingresamos, incluso la recta es la misma que presenta la figura 1.4 por lo que agregar estos pesos es de mucha ayuda.

2. Problema 2

En el siguiente ejercicio se nos solicita trabajar con los datos de $MNIST$ de dígitos escritos a mano de 28x28 pixeles. Se observa que hay un total de 10 grupos de datos ya que tenemos $K = 0, 1, ..., 9$, en la figura 2.1 se

ilustra como son los distintos tipos de datos.

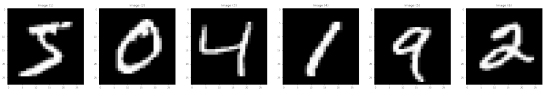

Figura 2.1: Dígitos escritos a mano.

Inciso a

El inciso nos plantea como constuir nuestra matriz Y, en este caso es una matriz de dimensión (nx$|K|$), donde n es el número de registros o datos que vamos a tomar y $|K|$ es el total de números que estamos ocupando, en este caso son: 0,1,2,..,9. La matriz Y queda de la siguiente forma:

$$Y = \begin{pmatrix} 0 & 0 & ... & 1 & ... & 0 \\ 0 & 0 & ... & 0 & ... & 1 \\ 0 & 0 & ... & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & ... & 0 & ... & 0 \end{pmatrix}$$

La matriz Y en los renglones cuenta con puros ceros y solo un valor 1 en la entrada correspondiente al digito que tienen los datos. Una vez que tenemos nuestras matrices(Y y la matriz de datos), lo que nos pide el ejercicio es realizar regresión multivariada pero tambien nos menciona que utilicemos dos conjuntos, uno correspondiente a los datos de entrenamiento y otro correspondiente a los datos de prueba. En este caso tomamos para el conjunto de entrenamiento cerca del 85 % y para el conjunto de prueba los restantes que es casi el 15 %.
Para realizar la regresión multivariada nos apoyamos de un lenguaje de programación(PYTHON). Ya que encontramos la $\hat{Y}$ observamos que no es muy claro a que grupo de los digitos corresponde cada registro o dato,

por lo que vamos a ocupar lo siguiente:

$$\hat{C}(x) = arg\ max_{k \epsilon K} \hat{y}_k.$$

Lo anterior nos va a ayudar a encontrar el valor máximo que tengamos en cada uno de los renglones de la matriz $\hat{Y}$ y una vez que encontramos este valor máximo se observe en que columna esta ubicado y de acuerdo a eso vamos a asignarle a que grupo de los digitos pertenece.
Al realizar dicho proceso nos falta observar que tan bueno fue el modelo que se ocupo, por lo que se hace uso de métricas de evaluación. En este caso vamos a usar las siguientes métricas: Presición, Recall, la medida F1 y la proporción de datos clasificados correctamente.
La explicación de las métricas y sus expresiones se anexan al final en un apendice por si existe duda.
Cabe recalcar que primero trabajamos con los datos de entrenamiento para despues ocupar los datos del conjunto de prueba y finalmente observar que tan bueno fue el modelo.

En la siguiente tabla observamos que nuestros valores para las distintas métricas, por ejemplo el valor correspondiente a Precisión se utiliza principalmente para medir que tan efectivo es el modelo en detectar la categoria de interes, en este caso observamos valores altos en su mayoria.
Para la recall se observa que tambien hay valores cercanos a 1 por lo que podemos entender que los datos de la categoria fueron bien clasificados.
Finalmente para f1-score encontramos que es una buena medida para resumir la evaluación en un solo número.

	precision	recall	f1-score	support
0	0.95	0.92	0.93	980
1	0.95	0.86	0.90	1135
2	0.91	0.76	0.83	1032
3	0.79	0.86	0.83	1010
4	0.93	0.74	0.82	982
5	0.95	0.45	0.61	892

6	0.88	0.91	0.89	958
7	0.92	0.81	0.86	1028
8	0.48	0.93	0.64	974
9	0.78	0.83	0.81	1009
accuracy			0.81	10000
macro avg	0.85	0.81	0.81	10000
weighted avg	0.85	0.81	0.82	10000

Tambien la proporción de datos clasificados correctamente fue de 0.81. En la figura 2.2 se ilustra mejor como el modelo de Baseline nos acomodo los datos en el cluster correspondiente.

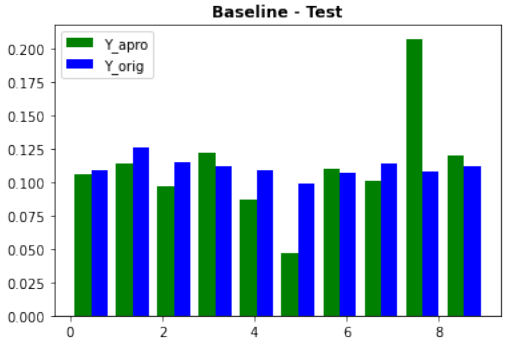

Figura 2.2: Baseline con datos de prueba.

El modelo fue bastante bueno ya que nos regreso buenos resultados en las métricas y al observar la figura 2.2 podemos ver que tan bueno fue al momento de realizar la selección del grupo.

Como dato adicional se agrega la matriz de confunción, donde la intención principal es identificar aquellos datos que fueron colocados en el lugar incorrecto, es decir, en la matriz que tenemos abajo si tomamos el valor correspondiente al renglón 1 y columna 3, vemos que ese valor nos dice

cuantos valores de la categoria cero, en este caso, estan clasificados en la categoria 2. En la diagonal se encuentra el número de datos que el modelo identifico en la categoria correcta.

$$MC = \begin{pmatrix} 903 & 0 & 4 & 4 & 0 & 6 & 22 & 1 & 38 & 2 \\ 0 & 971 & 4 & 9 & 1 & 0 & 5 & 0 & 145 & 0 \\ 10 & 6 & 789 & 35 & 7 & 0 & 36 & 8 & 134 & 7 \\ 4 & 1 & 17 & 871 & 0 & 0 & 8 & 13 & 86 & 10 \\ 1 & 12 & 11 & 6 & 727 & 0 & 13 & 1 & 119 & 92 \\ 6 & 5 & 9 & 119 & 8 & 401 & 24 & 9 & 279 & 32 \\ 13 & 4 & 11 & 2 & 8 & 9 & 872 & 0 & 39 & 0 \\ 4 & 18 & 12 & 16 & 14 & 0 & 3 & 834 & 42 & 85 \\ 5 & 5 & 6 & 20 & 4 & 5 & 10 & 3 & 904 & 12 \\ 8 & 3 & 7 & 16 & 14 & 0 & 0 & 39 & 80 & 842 \end{pmatrix}$$

Como principal intención de la matriz es que en las entradas distintas a la diagonal principal tengan cero, ya que eso no daria a entender que nuestro modelo esta haciendo un buen trabajo.

Inciso b

Para el siguiente inciso nos piden trabajar con LDA, QDA y regresión logística con los mismos datos y exponer los resultados. En este inciso vamos a ocupar las mismas métricas para avaluar cada uno de los métodos y decir cual fue el mejor.

Con el primer método que vamos a trabajar es con LDA, en este caso obtuvimos los siguientes valores:

	precision	recall	f1-score	support
0	0.94	0.96	0.95	980
1	0.89	0.97	0.93	1135
2	0.92	0.79	0.85	1032
3	0.87	0.87	0.87	1010

4	0.84	0.90	0.87	982
5	0.84	0.82	0.83	892
6	0.91	0.89	0.90	958
7	0.91	0.84	0.88	1028
8	0.80	0.81	0.80	974
9	0.81	0.85	0.83	1009
accuracy			0.87	10000
macro avg	0.87	0.87	0.87	10000
weighted avg	0.87	0.87	0.87	10000

Se observa que cada una de las métricas son buenas y si prestamos un poco más de atención son parecidos los resultados a Baseline, la proporción de datos clasificados correctamente fue de 0.87 por lo que le gane ligeramente a Baseline.
En la figura 2.3 observamos un comportamiento de los Y_apro más aproximado con ligeras diferencias respecto a los Y_orig.

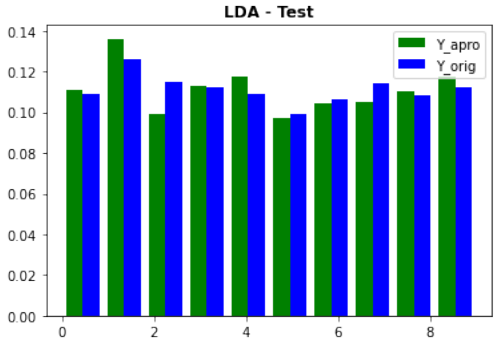

Figura 2.3: LDA con datos de prueba.

Igual agregamos la matriz de confunción y observamos que muchos de

nuestros datos fuera de la diagonal principal son distintos de cero por lo que no podemos tomar un buen juicio apartir de ella.

$$MC = \begin{pmatrix} 5586 & 4 & 21 & 32 & 23 & 103 & 52 & 2 & 91 & 9 \\ 0 & 6460 & 37 & 20 & 10 & 38 & 6 & 8 & 152 & 11 \\ 58 & 198 & 4862 & 180 & 117 & 25 & 189 & 42 & 256 & 31 \\ 14 & 93 & 159 & 5195 & 25 & 224 & 21 & 90 & 172 & 138 \\ 5 & 56 & 30 & 2 & 5243 & 44 & 29 & 2 & 54 & 377 \\ 57 & 59 & 26 & 247 & 50 & 4466 & 112 & 29 & 239 & 136 \\ 59 & 54 & 60 & 3 & 81 & 140 & 5437 & 0 & 80 & 4 \\ 35 & 143 & 38 & 46 & 180 & 14 & 1 & 5232 & 28 & 548 \\ 32 & 334 & 45 & 192 & 75 & 271 & 29 & 13 & 4697 & 163 \\ 32 & 27 & 16 & 94 & 319 & 27 & 0 & 266 & 57 & 5111 \end{pmatrix}$$

El siguiente método es QDA, en este método observamos la proporción de datos clasificados correctamente fue de 0.53, claramente es un valor pequeño comparado con el método de Basile y LDA, por lo que no nos ayudaria mucho ocuparlo.

	precision	recall	f1-score	support
0	0.35	0.97	0.52	980
1	0.90	0.95	0.92	1135
2	0.91	0.16	0.27	1032
3	0.63	0.25	0.35	1010
4	0.92	0.06	0.12	982
5	0.68	0.03	0.06	892
6	0.69	0.96	0.80	958
7	0.90	0.27	0.41	1028
8	0.42	0.65	0.51	974
9	0.42	0.95	0.58	1009
accuracy			0.53	10000
macro avg	0.68	0.52	0.45	10000

```
weighted avg       0.69      0.53      0.46      10000
```

En la figura 2.4 es todavía más claro que el método tiene muchas fallas ya que la asignación de las categorias no las hace de forma correcta.

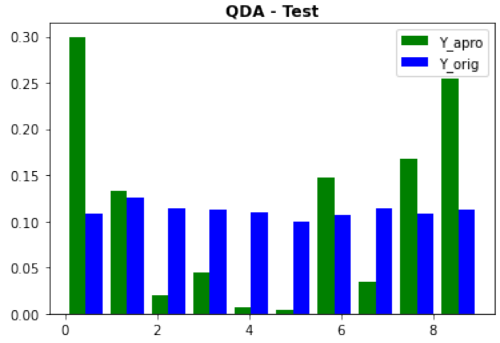

Figura 2.4: QDA con datos de prueba.

En cuanto a la matriz de confunción observamos que muchas entradas fuera de la diagonal principal son bastante altas por lo que podría ser un buen indicio de que el método no es tan bueno para estos datos en particular.

$$MC = \begin{pmatrix} 949 & 1 & 0 & 6 & 0 & 2 & 9 & 1 & 11 & 1 \\ 16 & 1075 & 2 & 1 & 0 & 0 & 18 & 0 & 18 & 5 \\ 485 & 10 & 161 & 91 & 2 & 0 & 184 & 3 & 84 & 12 \\ 473 & 15 & 1 & 248 & 0 & 0 & 37 & 3 & 159 & 74 \\ 236 & 6 & 4 & 10 & 61 & 2 & 60 & 11 & 208 & 384 \\ 348 & 10 & 1 & 13 & 0 & 27 & 67 & 2 & 355 & 69 \\ 24 & 4 & 2 & 1 & 0 & 3 & 916 & 0 & 8 & 0 \\ 21 & 8 & 2 & 13 & 1 & 0 & 3 & 275 & 25 & 680 \\ 127 & 63 & 2 & 10 & 0 & 6 & 26 & 3 & 637 & 100 \\ 19 & 9 & 2 & 3 & 2 & 0 & 1 & 8 & 6 & 959 \end{pmatrix}$$

Finalmente para regresión logistica tenemos una propocion de datos clasificados correctamente del 0.92 que muy buena comparada con los anteriores. Se observa que en muchas entradas de los valores de la métricas los valores son altos y eso no ayuda bastante a darnos una buena idea del desempeño de Regresión Logística.

	precision	recall	f1-score	support
0	0.95	0.96	0.95	980
1	0.95	0.98	0.96	1135
2	0.91	0.89	0.90	1032
3	0.90	0.91	0.91	1010
4	0.94	0.93	0.93	982
5	0.89	0.87	0.88	892
6	0.94	0.94	0.94	958
7	0.93	0.92	0.92	1028
8	0.88	0.88	0.88	974
9	0.91	0.92	0.91	1009
accuracy			0.92	10000
macro avg	0.92	0.92	0.92	10000
weighted avg	0.92	0.92	0.92	10000

En la figura 2.5 observamos un comportamiento muy cercado por parte de Y_apro, ya que las barras de las gráficas se observan muy pecadas una de la otra.

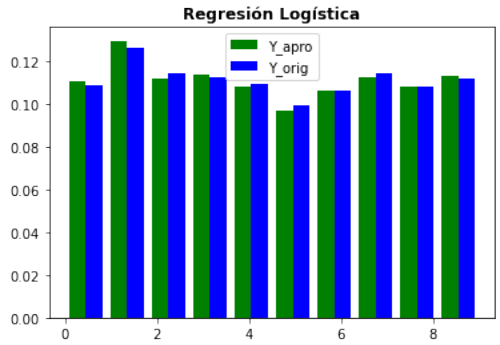

Figura 2.5: Regresión Logística con datos de prueba.

En la matriz de confunsión observamos valores fuera de la diagonal pequeños y a simple vista podría ser buen inicio de que el método es bueno.

$$MC = \begin{pmatrix} 942 & 0 & 7 & 3 & 1 & 15 & 5 & 4 & 3 & 0 \\ 0 & 1107 & 8 & 3 & 0 & 2 & 3 & 2 & 10 & 0 \\ 12 & 13 & 915 & 19 & 11 & 5 & 11 & 7 & 34 & 5 \\ 3 & 2 & 21 & 920 & 2 & 21 & 1 & 12 & 19 & 9 \\ 2 & 3 & 11 & 3 & 909 & 1 & 8 & 6 & 7 & 32 \\ 9 & 3 & 3 & 35 & 9 & 775 & 15 & 6 & 32 & 5 \\ 9 & 4 & 13 & 2 & 7 & 18 & 904 & 0 & 1 & 0 \\ 1 & 12 & 24 & 6 & 6 & 2 & 0 & 943 & 6 & 28 \\ 10 & 14 & 5 & 23 & 8 & 27 & 9 & 8 & 855 & 15 \\ 6 & 8 & 2 & 9 & 19 & 8 & 1 & 23 & 8 & 925 \end{pmatrix}$$

A manera de conclusión podemos decir que los métodos que mejor se comportaron fueron Baseline, LDA y Regresión Logística. Entre esos tres métodos encontramos buenos valores de las métricas incluso entre estos tres métodos el mejor es Regresión Logística ya que obtuvimos una proporción de datos clasificados correctamente de 0.92, que con ese valor nos dice que este método es muy bueno para tratar nuestros datos.

Como se observo en la figura 2.1 las imagenes con las que trabajamos cuenta con un tamaño de 28x28 pixeles por lo que si obsevamos las imagenes detalladamente estan ligeramente borrosas por lo que esta parte podría ser un factor importante que posiblemente ayude a mejorar la asignación de los datos a su categoria o grupo respectivamente, en cuanto a la cantidad de datos no hay mucho que decir debido a que si son suficiente para trabajar con ellos.

3. Problema 3

En el siguiente ejercicio nos piden considerar un conjunto de datos que corresponden a opiniones de usuarios en los siguientes productos: automóviles, hoteles, lavadoras, libros, teléfonos celulares, música, computadoras y películas. Además no agregan que cada opinion una categoria correspondiente a un sentimiento positivo(yes) o negativo(no).
Como primer paso a los textos se le realizo un preproceso donde eliminamos muchos detalles que quiza no nos ayuden a nuestro analisis incluso lo vuelvan más complicado. Entre las medidas que se tomaron fue eliminar acentos, remover signos o elementos raros, quitar conectores y decidir si trabajar los datos en mayusculas o minúsculas.

Inciso a

En el siguiente inciso nos piden realizar una representación vectorial de los textos mediante Bag Of Words y para poder realizar una buena representación hemos tomado en cuenta unos criterios importantes como lo es la frecuencia con la que algunos de nuestras palabras se repiden y tambien el volumen de palabras que vamos a tomar en cuenta. Al hablar del volumen de palabras que vamos a tomar en cuenta nos referimos a la cantidad de palabras que tomamos del diccionario y con ellas vamos a ir texto por texto contando cuantas de ellas aparecen en dicho texto, gracias a esta idea vamos a poder representar los textos en forma vectorial. Como primer factor antes de realizar la representación final fue tomar la frecuencia

con la que algunas palabras se repiten, en este caso tomamos la cantidad de 10,000.
En la figura 3.1 se ilustra la gráfica de la frecuencia acumulada por parte de las 10,000 palabras que tomamos al principio y nos indica que tomemos al menos 4,000 de ellas para tener el 80 % de las palabras que más se repitieron.

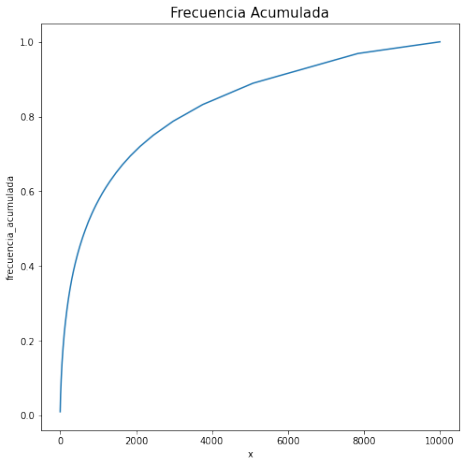

Figura 3.1: Gráfico de frecuencia acumulada de las letras.

Como resultado de realizar el proceso y tomar en cuenta aquellas palabras que se repiten un número considerado de veces tenemos como resultado la siguiente tabla, solo con la intención de ilustrar esta representación se muestra la vectorización que se le realizo a cada texto.
La reducción de dimensión de nuestra representación vectorial tiene como fin no trabajar los datos en tan alta dimensión como en un principio los teniamos.

Inciso b

	si	pelicula	solo	asi	bien	..	legal
0	3	0	0	1	0	...	0
1	8	0	0	6	4	...	0
2	0	1	0	0	2	...	0
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
398	0	0	0	0	1	...	0
399	0	1	0	0	0	...	0

Cuadro 3.1: Fruta disponible

En este inciso nos piden encontrar una matriz de similaridades de los textos usando la representación BOW y la distancia del coseno para ayudarnos a encontrar patrones. En la siguiente matriz se ilustra el resultado de encontrar dicha matriz, en este caso contamos con una matriz simetrica de 400x400 ya que tomamos todos los textos disponibles.

$$matriz\ similaridades = \begin{pmatrix} 1{,}0 & 0{,}31 & ... & 0{,}125 & ... & 0{,}057 \\ 0{,}31 & 1{,}0 & ... & 0 & ... & 0{,}036189 \\ 0{,}087 & 0{,}06 & ... & 1 & ... & 0{,}0865 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0{,}057 & 0{,}03618 & ... & 0{,}022 & ... & 1{,}0 \end{pmatrix}$$

Una vez que ya tenemos nuestros datos representados por la matriz de similaridades intentamos encontrar una buena representación, donde a lo mejor encontremos los datos separados y agrupados en su respectiva categoria, note que en este ejercicio se esta tomando en cuenta dos etiquetas que son: categoria y sentimiento.
Para encontrar patrones interesante nos basamos en la representación que nos puede proporcionar Kernel PCA.

En la figura 3.2 tenemos la representación en el cuarto y tercer componente

principal al aplicar PCA tomando en cuenta el sentimiento que cada texto tiene. A simple vista no observamos un buen patrón pero a lo mejor una forma curiosa de como proyecta los datos pero no los separa como quisieramos. Los parámetros que ocupamos para encontrar esta representación fue un kernel cosine y pedimos un total de 6 componentes principales.

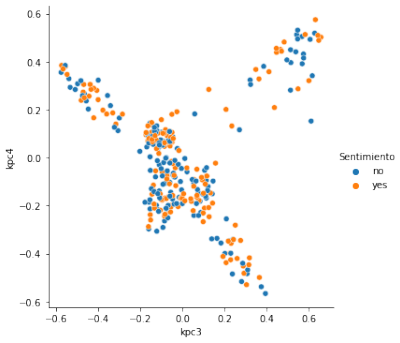

Figura 3.2: Representación con kernel PCA(componente 3 y componente 4, sentimiento).

En la figura 3.3 cambiamos el sentimiento por las categorias que se mencionaron al principio y en este caso si obsevamos buenos patrones. Los parámetros que ocupamos fueron los mismo solo se cambio el sentimiento por las categorias.

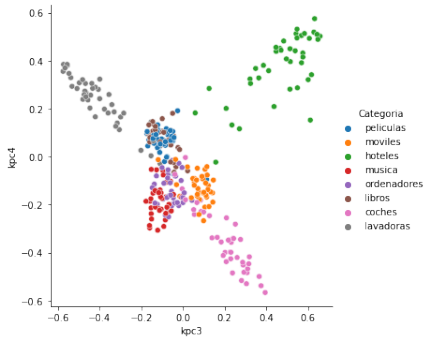

Figura 3.3: Representación con kernel PCA(componente 3 y componente 4, categoria).

En esta representación encontramos un buen patrón ya que logra juntar datos de una misma categoria, por ejemplo, observamos que los textos de la categoria hoteles los manda a una punta, lo mismo con los textos de lavadoras y coches. A primer instancia podemos inferir que quiza estos textos tengas ciertas palabras que no comparten con la mayoria de los textos y estas pequeñas diferencias quiza nos ayudaron a encontrar esta represntación. Los datos que no se pudieron separar fueron los que hablan de música, moviles, ordenadores,libros y peliculas, puede que al menos en esta representación no se lograron separar lo sufiente pero como tenemos 6 representaciones distintas en muchas de ellas si se logran separar.
Por ejemplo en la figura 3.4 observamos que nos separa bien los textos que hablan de música y moviles, esto no ocurrio en la figura 3.3 por lo que estas representaciones son buenas, no se agregaron todos los gráficos pero al analizarlo encontramos que en muchos de ellos no ayudan a separar cierto grupo de textos por lo que esta representación es adecuada para los datos al menos para las categorias.

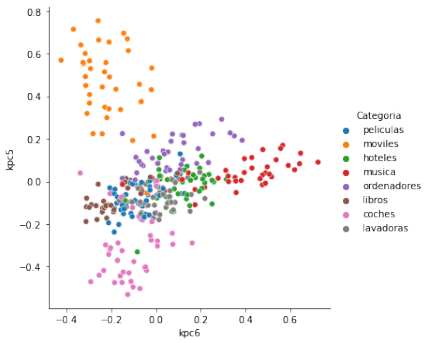

Figura 3.4: Representación con kernel PCA(componente 6 y componente 5, categoria).

En la figura 3.5 introducimos otro tipo de kernel, en este caso fue un kernel polynomial con un sigma igual a 2.05. Al menos para el sentimiento no encontramos un buen patrón.

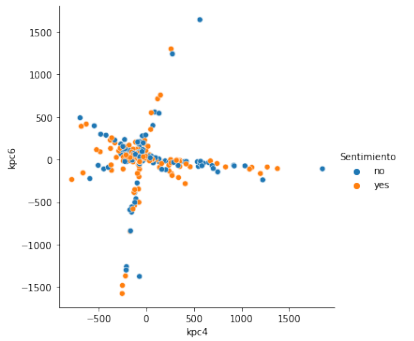

Figura 3.5: Representación con kernel PCA(componente 4 y componente 6, sentimiento).

En la figura 3.6 encontramos un buen patrón por parte de los datos. Observamos que textos que hablan de hoteles, libros, música, moviles, peliculas

y labadoras nos lo manda a distintas puntas del gráfico. Los parámetros que se ocuparon fueron los mismo que en la figura 3.5.

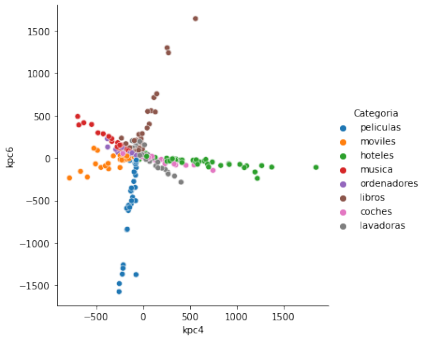

Figura 3.6: Representación con kernel PCA(componente 4 y componente 6, categoria).

Finalmente el método que nos dio los mejores resultados fue T-SNE, en este caso tomamos la matriz que resulto del BOW y las distancias coseno, el valor del parámetro de perplejidad fue de 10.
En la figura 3.7 obsevamos que nos ayuda a agrupar muy bien nuestros datos y no solo juntos los textos de la misma categoria si no que los pone los pone alejados entre ellos.

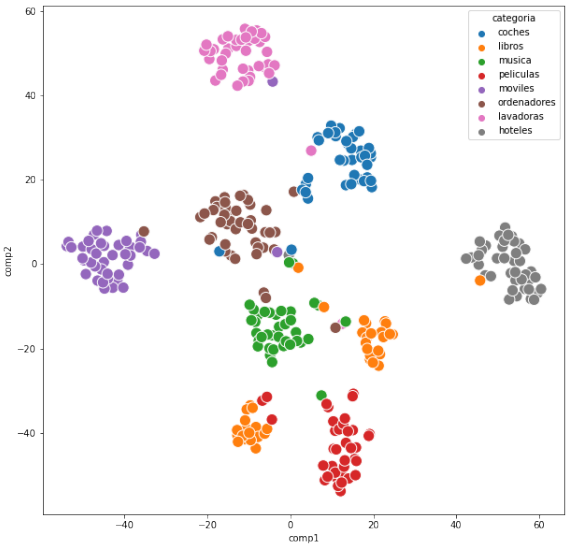

Figura 3.7: T-SNE.

Observando ambos tipos de kernel se obtuvo buenos resultados pero solo en la categorias ya que en el sentimiento los sigue revolviendo un poco los registros. Las representaciones que se obtuvieron fueron muy buenas para ambos tipos de kernel pero el kernel cosine nos mostro una mejor representación con otros componentes principales que no mostramos pero si nos proporciona buenos resultados.
Para spectral se encontro muy buena representación de los datos, incluso mejor que con los anteriores métodos ya que no solo los agrupos también los alejos entre ellos.

Inciso c

En este inciso nos piden ocupar distintos métodos de clustering para tratar de indentificar las categorias de los textos o sentimiento. En este inciso

vamos a ocupar las representaciones que encontramos en el inciso b) ya que muchas de ellas nos proporcionan buenos valores para ciertas caracteristicas que hacen diferente un texto del otro.
Como primer método de clustering ocupamos K-means. En la figura 3.8 tomamos la representación que encontramos con el kernel cosine con los 6 componentes principales. En este caso vemos cierto parecido con los anteriores gráficos por lo que al menos para las categorias que ya mencionamos si encontramos una buena selección.

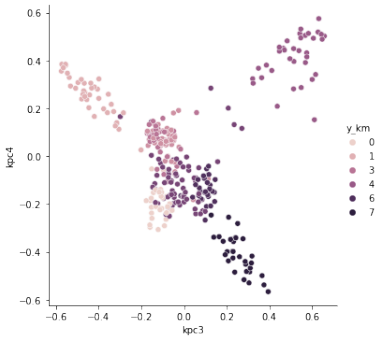

Figura 3.8: Método de k-means(Representación kernelPCA-cosine).

En la figura 3.9 ahora tomamos la representación con kernel polynomial y los 6 componentes principales. Si obsevamos hay cierta similitud pero no es muy bueno el método para esta representación ya que confunde dos grupos de textos.

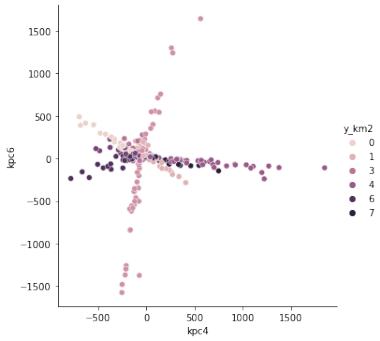

Figura 3.9: Método de k-means(Representación kernelPCA-poly).

El siguiente método que ocupamos fue Fuzzy-kmeans, con este método intentamos encontrar un patrón cercano a lo que nos resulto en la figura 3.4 tomando la representación que en los 6 componentes principales con un kernel cosine. En la figura 3.10 notamos que no es muy buena la representación por lo que no nos ayuda mucho.

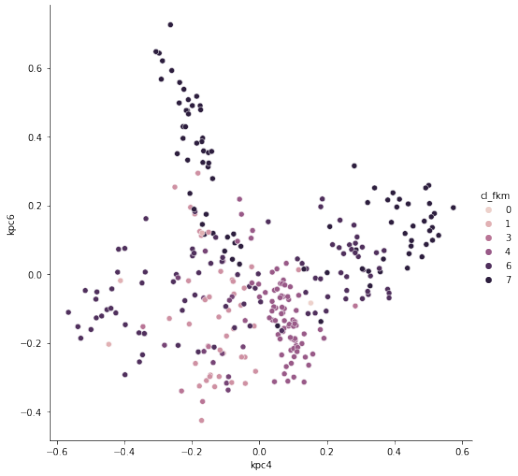

Figura 3.10: Método de Fuzzy-kmeans(Representación kernelPCA-cosine).

En la figura 3.11 realizamos el mismo método pero ahora tomamos en cuenta el sentimiento de cada texto, el resultado no es nada bueno comparado con el visto previamente.

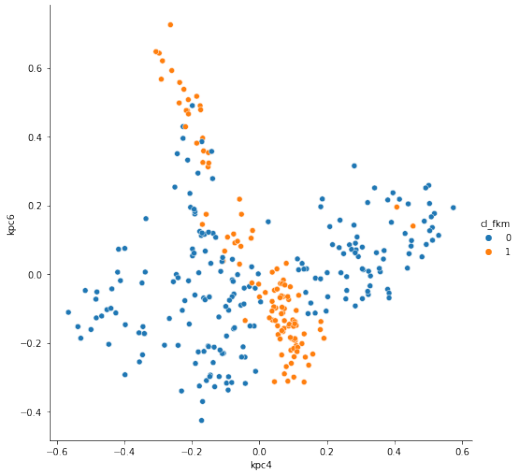

Figura 3.11: Método de Fuzzy-kmeans(Representación kernelPCA-cosine-sentimiento).

Con la representación del kernel polynomial no se agrego debido a que no hubo mucha mejoria comparada con el resultado de k-means.

Otro método interesante es Spectral Embedding, con este método logramos buenos resultados parecidos a los de kernel PCA.
En la figura 3.12 observamos que nuestros datos se logran separar bien pero no como quisieramos, en este caso estamos tomando en cuenta la representación con kernel PCA con un kernel cosine.

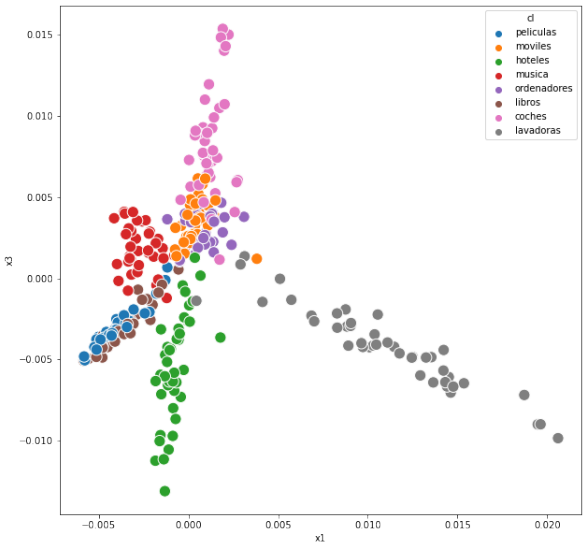

Figura 3.12: Spectral Embddinng(componente 1 y 3).

En este caso se omitieron las posibles representaciones que resultaran de T-SNE ya que no habia mucha diferencia respecto a kernel PCA incluso no hubo mucha mejoria, prueba de eso tenemos la figura 3.13 que no logra proporcionarnos buenos grupos.

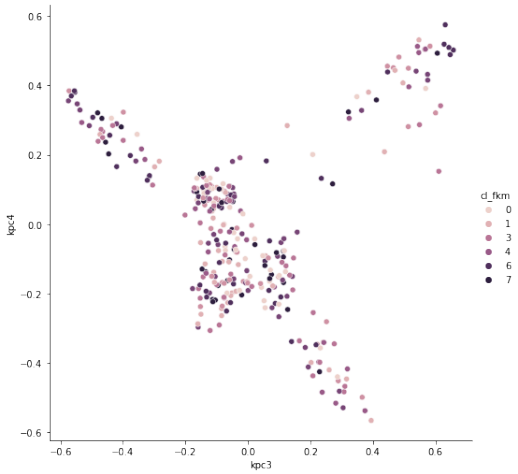

Figura 3.13: Método de Fuzzy-kmeans(Representación T-SNE-categoria)

Inciso d

El inciso nos pide que basado en BOW y una buena representación de los datos utilicemos LDA, QDA y regresión Logística para clasificar el sentimiento de los textos y el tópico. Empezamos con LDA donde una vez que seleccionamos nuestros datos en una adecuada representación, que se menciono al principio del ejercicio y una división de los 400 textos, tomando 320 textos como entrenamiento y 80 como textos de prueba, nos dio los siguientes resultados. Primero vamos a mostrar los resultados que nos arrojo tomando en cuenta las categorias y las siguientes métricas: Presición, Recall, la medida F1 y la proporción de datos clasificados correctamente. En la siguiente tabla observamos los valores de las métricas, en este caso la proporción de datos clasificados correctamente es alto, es del 0.82 por lo que si nos ayuda a clasificar las categorias de manera adecuada.

```
              precision    recall  f1-score   support
```

coches	0.75	0.69	0.72	13
hoteles	1.00	1.00	1.00	13
lavadoras	0.91	0.91	0.91	11
libros	0.64	1.00	0.78	7
moviles	0.67	0.75	0.71	8
musica	0.91	1.00	0.95	10
ordenadores	0.67	0.80	0.73	5
peliculas	1.00	0.54	0.70	13
accuracy			0.82	80
macro avg	0.82	0.84	0.81	80
weighted avg	0.85	0.82	0.82	80

Para que se más claro ver los resultados en la figura 3.14 se muestran y no hay mucha diferencia entre ellos más que en peliculas.

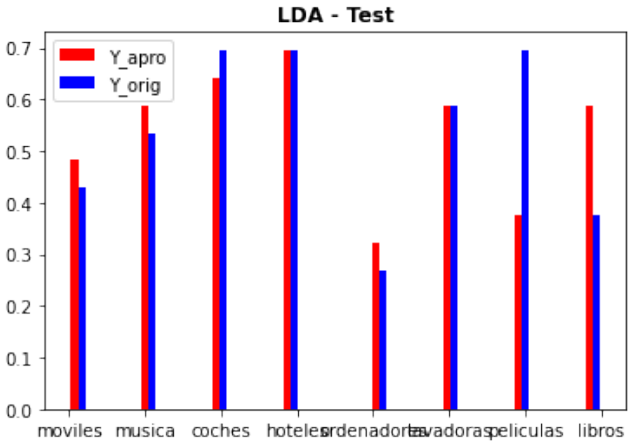

Figura 3.14: LDA - Categorias

El siguiente es QDA, donde seguimos evaluando las categorias. Los resultados al ocupar QDA fueron las siguientes:

	precision	recall	f1-score	support
coches	0.00	0.00	0.00	13
hoteles	0.00	0.00	0.00	13
lavadoras	0.50	0.09	0.15	11
libros	0.03	0.14	0.06	7
moviles	0.00	0.00	0.00	8
musica	0.07	0.10	0.08	10
ordenadores	0.00	0.00	0.00	5
peliculas	0.06	0.08	0.07	13
accuracy			0.05	80
macro avg	0.08	0.05	0.04	80
weighted avg	0.09	0.05	0.05	80

En la tabla de arriba observamos que nuestras metricas reflejan muy malos resultados por lo que no es muy confiable QDA para las categorias.

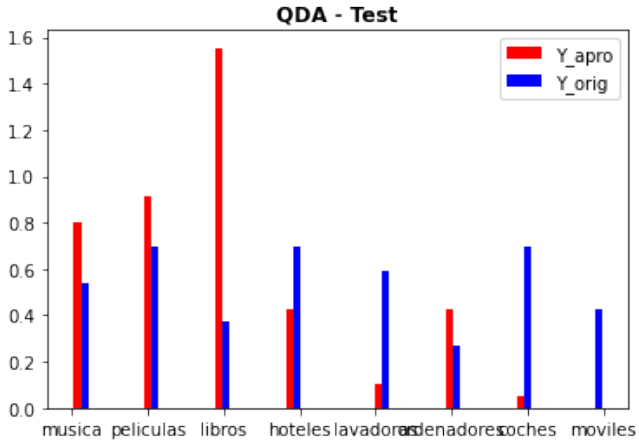

Figura 3.15: QDA - Categorias

En la figura 3.15 se refleja que no QDA no funciona bien para las categorias pero todavía no observamos para el sentimiento por lo que todavía no lo debemos descartar.

Con Regresión Logística tomando en cuenta las categorias, los resultados de las métricas fueron las siguientes:

	precision	recall	f1-score	support
coches	0.92	0.85	0.88	13
hoteles	1.00	1.00	1.00	13
lavadoras	1.00	0.91	0.95	11
libros	0.78	1.00	0.88	7
moviles	0.88	0.88	0.88	8
musica	0.71	1.00	0.83	10
ordenadores	0.83	1.00	0.91	5
peliculas	1.00	0.62	0.76	13
accuracy			0.89	80
macro avg	0.89	0.91	0.89	80

Al observar los valores de las métricas nos da buen indicio de que mejor mucho comparado con LDA y obviamente QDA. En este caso vemos que el proporcion de datos clasificados correctamente es del 0.89 cerca del 0.90 por lo que Regresión Logística es el mejor entre los tres para clasificar las categorias.

Figura 3.16: Regresión Logística - Categorias

En la figura 3.16 obsevamos que falla en ciertas categorias pero en otras no fallo en nada por lo que podriamos agregar que ayuda a identificar ciertas categorias entre las 8.

Realizaremos el mismo analisis pero ahora solo nos interesa ver que tan bueno es LDA, QDA y Regresión Logística para clasificar los textos por su sentimiento.
Comenzamos con LDA, el resultado de las metricas se muestran a continuación:

	precision	recall	f1-score	support
no	0.41	0.33	0.37	36
yes	0.53	0.61	0.57	44
accuracy			0.49	80
macro avg	0.47	0.47	0.47	80
weighted avg	0.48	0.49	0.48	80

Los resultados no fueron muy buenos y es fácil observar que no nos ayuda mucho. En la figura 2.1 ilustramos los resultados de realizar LDA.

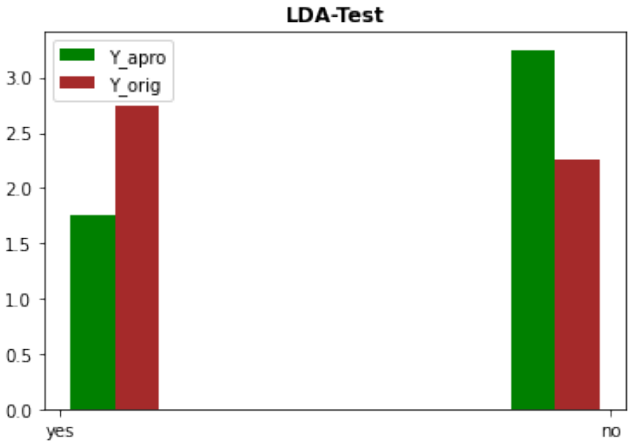

Figura 3.17: LDA - Sentimiento

Para QDA tenemos los siguientes resultados para las métricas fueron las siguientes:

	precision	recall	f1-score	support
no	0.41	0.39	0.40	36
yes	0.52	0.55	0.53	44
accuracy			0.48	80
macro avg	0.47	0.47	0.47	80
weighted avg	0.47	0.47	0.47	80

Los resultados no mejoran comparados con LDA y es fácil observalos en la figura 3.18.

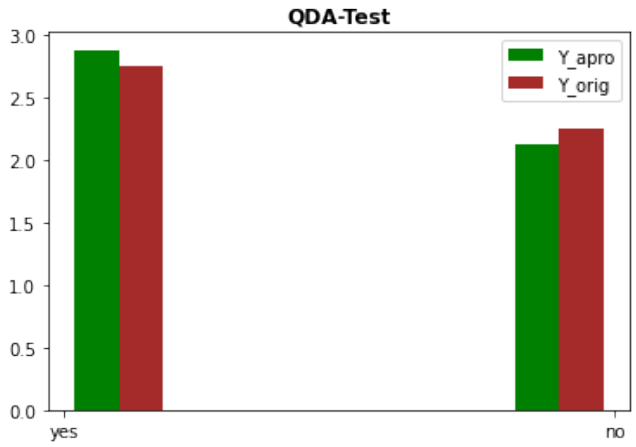

Figura 3.18: LDA - Sentimiento

Finalmente para Regresión Logística tenemos como resultados de las métricas:

	precision	recall	f1-score	support
no	0.62	0.81	0.70	36
yes	0.79	0.59	0.68	44
accuracy			0.69	80
macro avg	0.70	0.70	0.69	80
weighted avg	0.71	0.69	0.69	80

Con ayuda de la figura 3.19 y la anterior tabla encontramos los resultados de aplicar Regresión Logística y comparado con los anteriores podemos decir que es el mejor entre los tres.

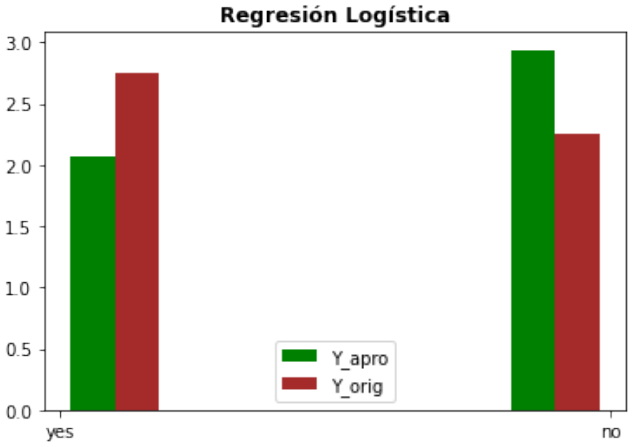

Figura 3.19: Regresión Logística - Sentimiento

Para ambos casos: Categorias y Sentimiento, el mejor de los tres fue regresión Logística ya que de acuerdo a las métricas tenemos mejores resultados en varios de ellos e incluso en los gráficos se observa mejoria. El que no ayudo mucho fue QDA ya que se equivoco mucho en la asignación de la categoria en ambos casos.