INF 393-Tarea 2

Métodos lineales para clasificación

Pregunta 1

Reducción de dimensionalidad para clasificación

Definición del problema

Reconocimiento de sonidos de vocales del idioma inglés.

Dataset generado mediante el registro de la pronunciación de las palabras por 15 personas.

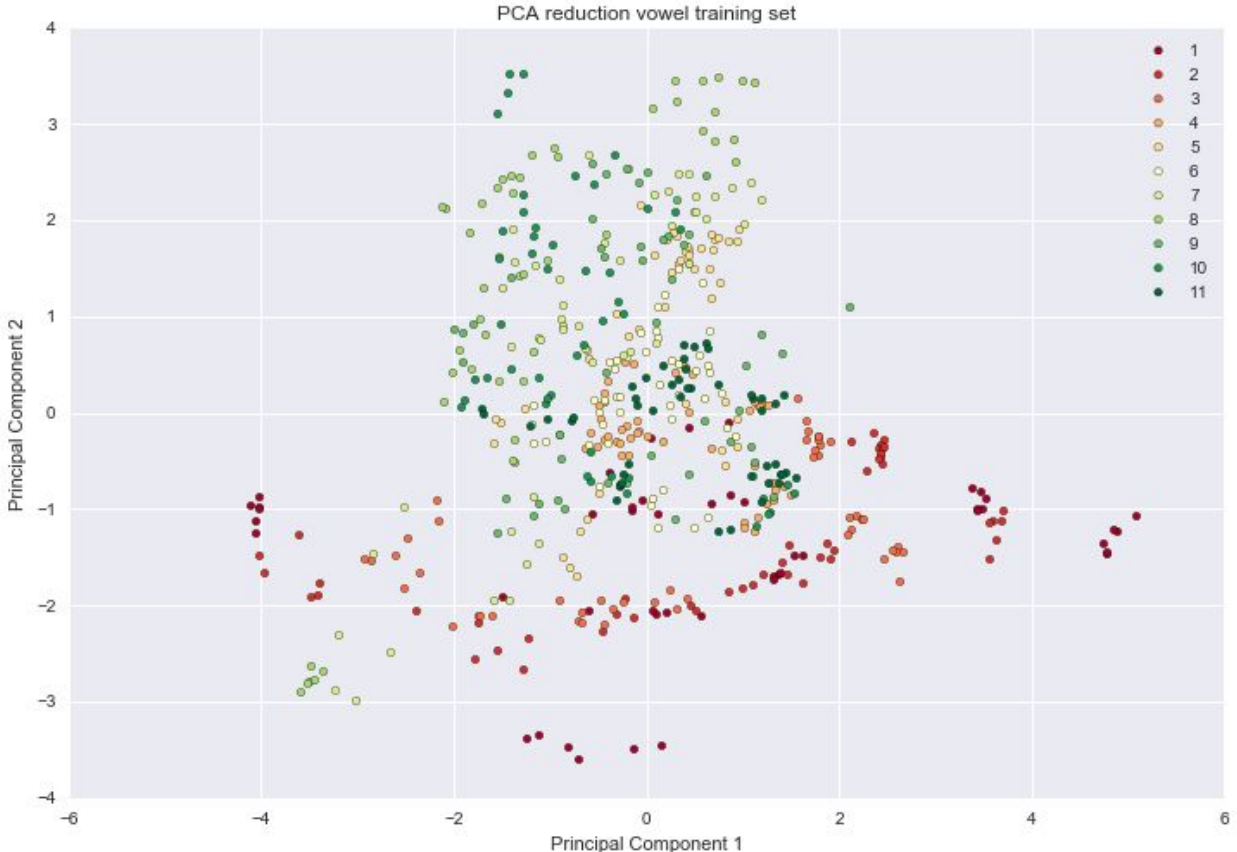
Vocal	Palabra	Vocal	Palabra
i	heed	o	hod
I	hid	C:	hoard
E	head	U	hood
A	had	u:	who'd
a:	hard	3:	heard
Y	hud		

El dataset cuenta con 10 features por cada ejemplo (Rabiner y Schaffer).

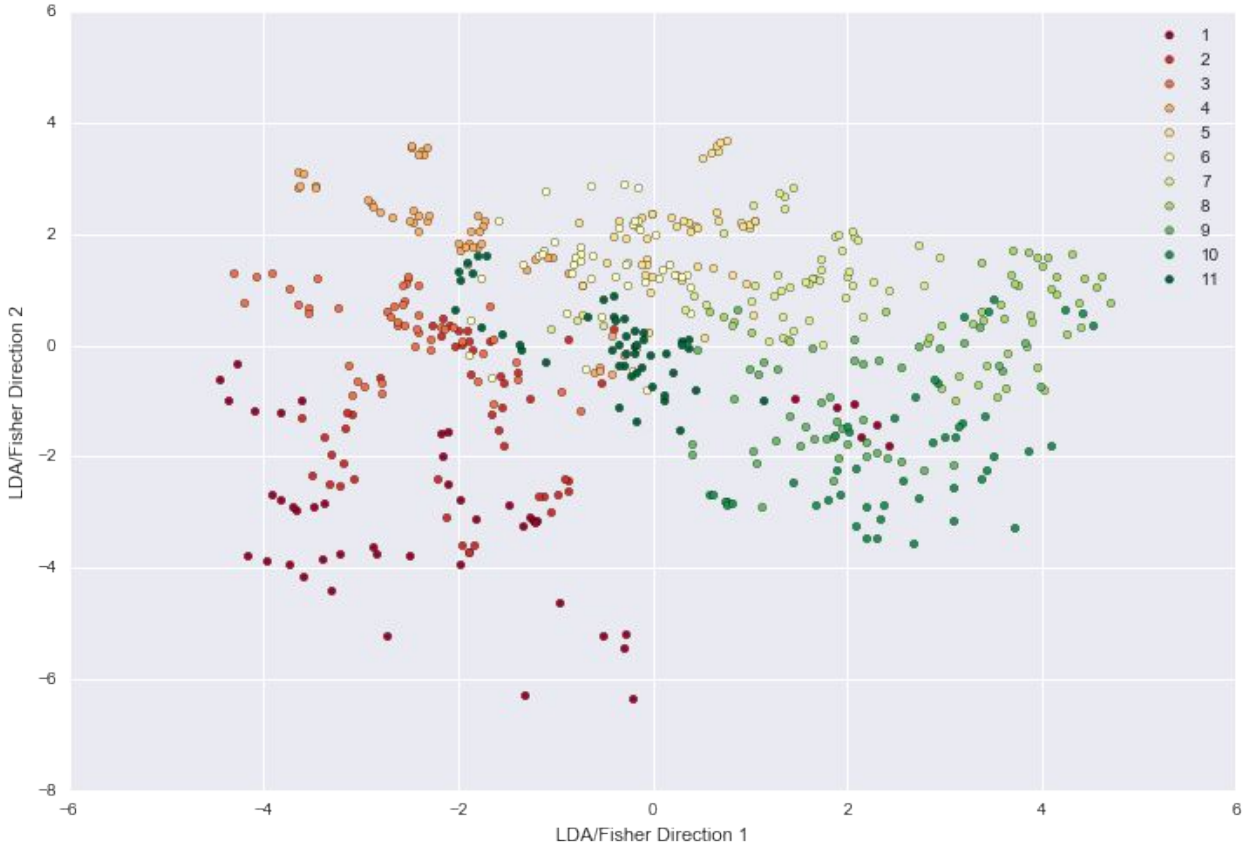
Reducción de dimensionalidad

En general cada reducción busca, en el mejor de los casos disminuir el ruido del modelo y además, eliminar datos que puedan ser redundantes dentro del problema estudiado. Además facilita la visualización de la data y ganancias en rendimiento.

Reducción de dimensionalidad vía PCA



Reducción de dimensionalidad vía PCA



Análisis cualitativo

Como se ha estudiado en clases LDA por lo general presenta mejores resultados que PCA, puesto que el primero toma en consideración las etiquetas para realizar la reducción

La proyección realizada por PCA no facilita la detección de áreas en las que sea predominante una clase sobre la otra. Las clases 6 y 8 particularmente están muy dispersas en el espacio 2D.

En cambio la proyección realizada por LDA tiende a ordenar de mejor manera los ejemplos. Las clases 4 y 3 tienden a agruparse mejor. Sin embargo, las clases 5 y 6 tienden a estar superpuestas (al igual que la 9 y 10)

Clasificador en base a probabilidad a priori

Se calculan las probabilidades a priori de cada clase. En base a ellas se construye un intervalo para cada clase, el cual indicará la pertenencia del ejemplo a dicha clase. El intervalo para la clase i se construye de la forma:

Intervalo de clase i = [Probabilidad acumulada i-1, Probabilidad acumulada i]

Al ejecutar el clasificador se escoge un número aleatorio entre 0 y 1, y en base al intervalo en el que se encuentra dicho número se obtiene la clase a la que pertenece.

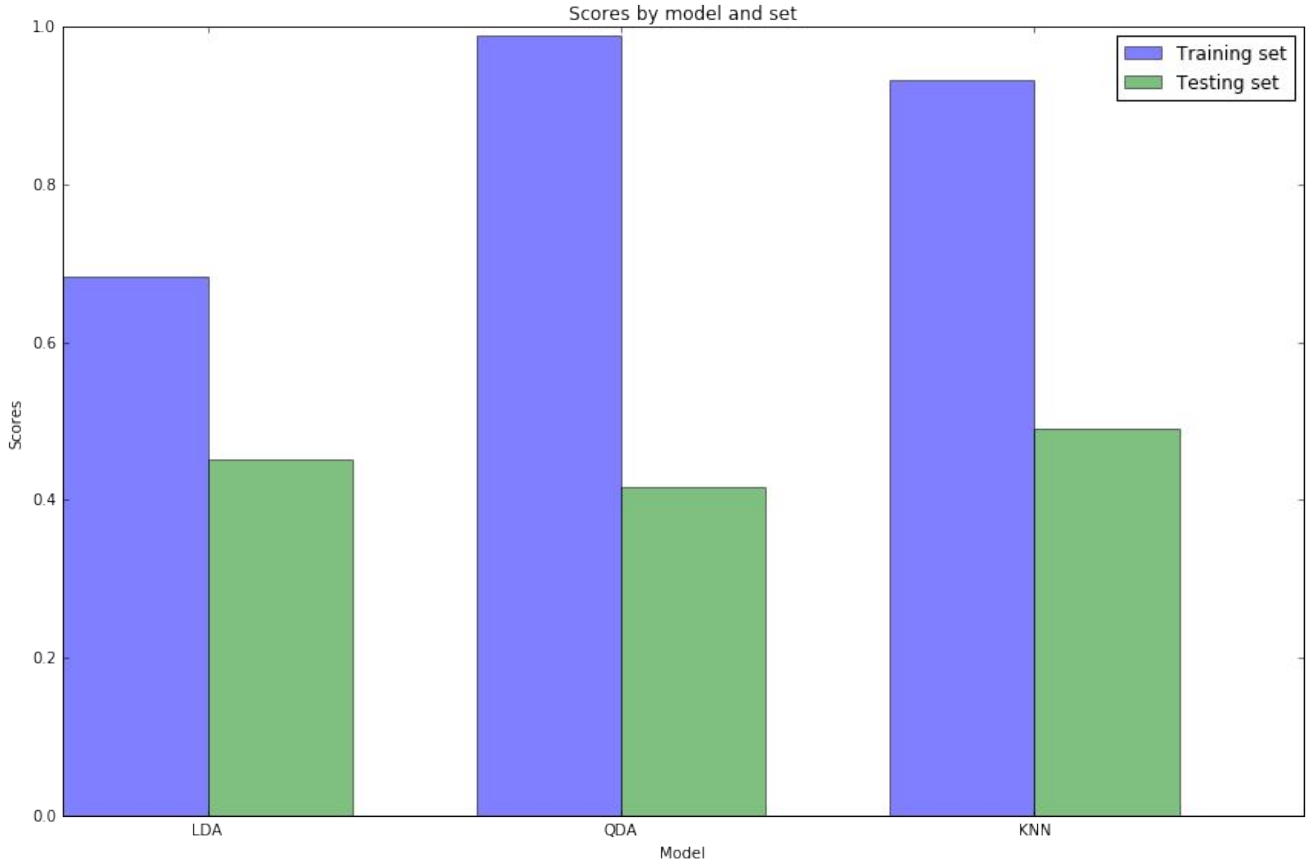
La precisión del modelo se obtuvo promediando la precisión obtenida en 1000 iteraciones sobre el dataset de test. Obteniendo un valor de 0.09

LDA vs QDA vs KNN
(sin reducir dimensiones)

El modelo LDA genera fronteras lineales para separar las clases, mientras que QDA genera fronteras cuadráticas, a cambio de un costo computacional más elevado.

KNN por otra parte es un clasificador no lineal que infiere la clase a la que pertenece un ejemplo en base a sus K vecinos más cercanos. La etiqueta es determinada en base a la que es más predominante entre sus vecinos.

Resultados


Scores by model and set
Training set
Testing set
1.0
0.8
0.6
0.4
0.2
0.0
Scores
LDA
QDA
KNN
Model

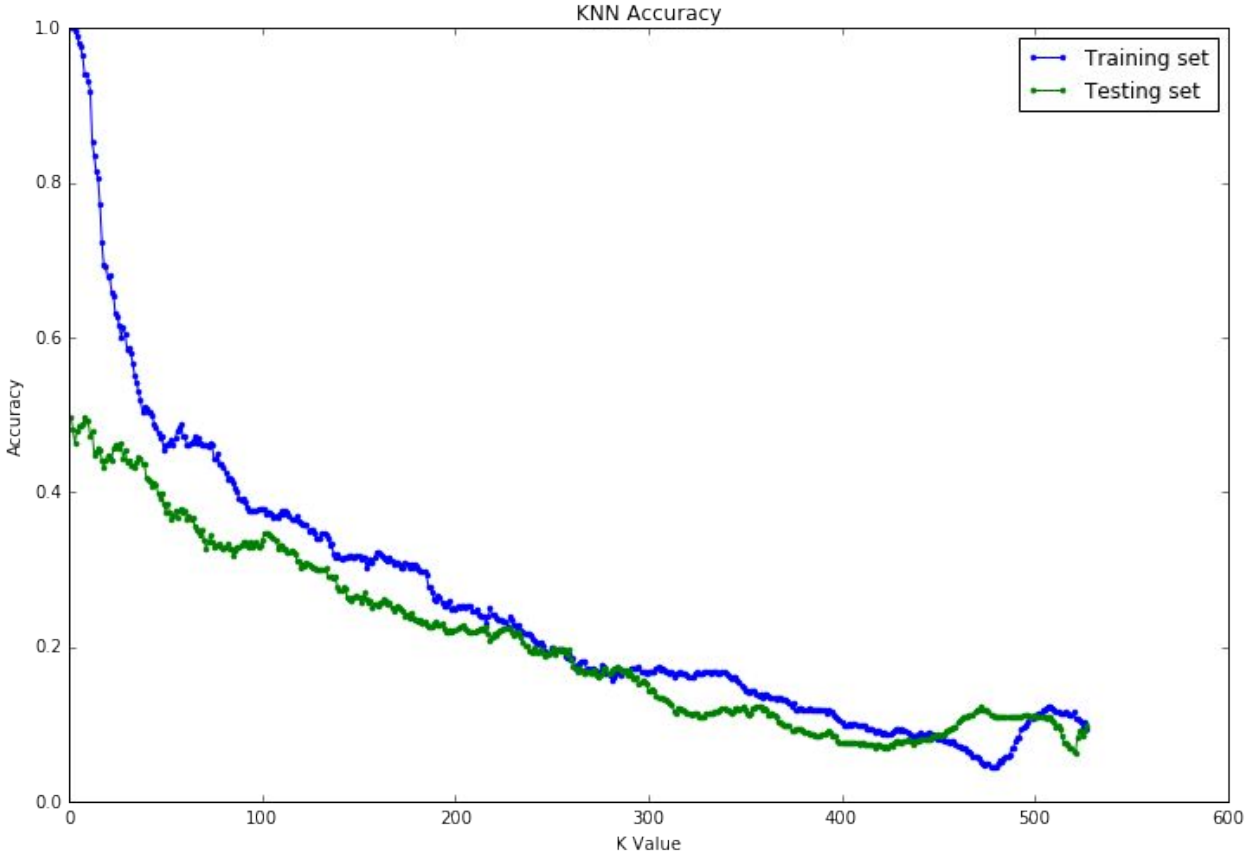
Overfitting?

En general todos los modelos tienden a sobre-ajustarse al set de entrenamiento. Los autores encontraron la mejor precisión para el modelo de Gaussian node network y Square node Network con un 55% de precisión (KNN logra un 49%). Los autores proponen un aumento del tamaño del dataset mediante la inclusión de grabaciones realizadas por una mayor cantidad de personas para mejorar esta situación.

Variación de K para el modelo de KNN

Al variar K se varía la cantidad de vecinos que son considerados al momento de inferir la clase a la que pertenece un ejemplo. Los vecinos más cercanos son escogidos por lo general utilizando la distancia euclideana. Cuando K es igual 1 entonces el modelo se conoce como “vecino más cercano”.

Resultados K 1~528



Análisis Cualitativo

Variación de K para KNN

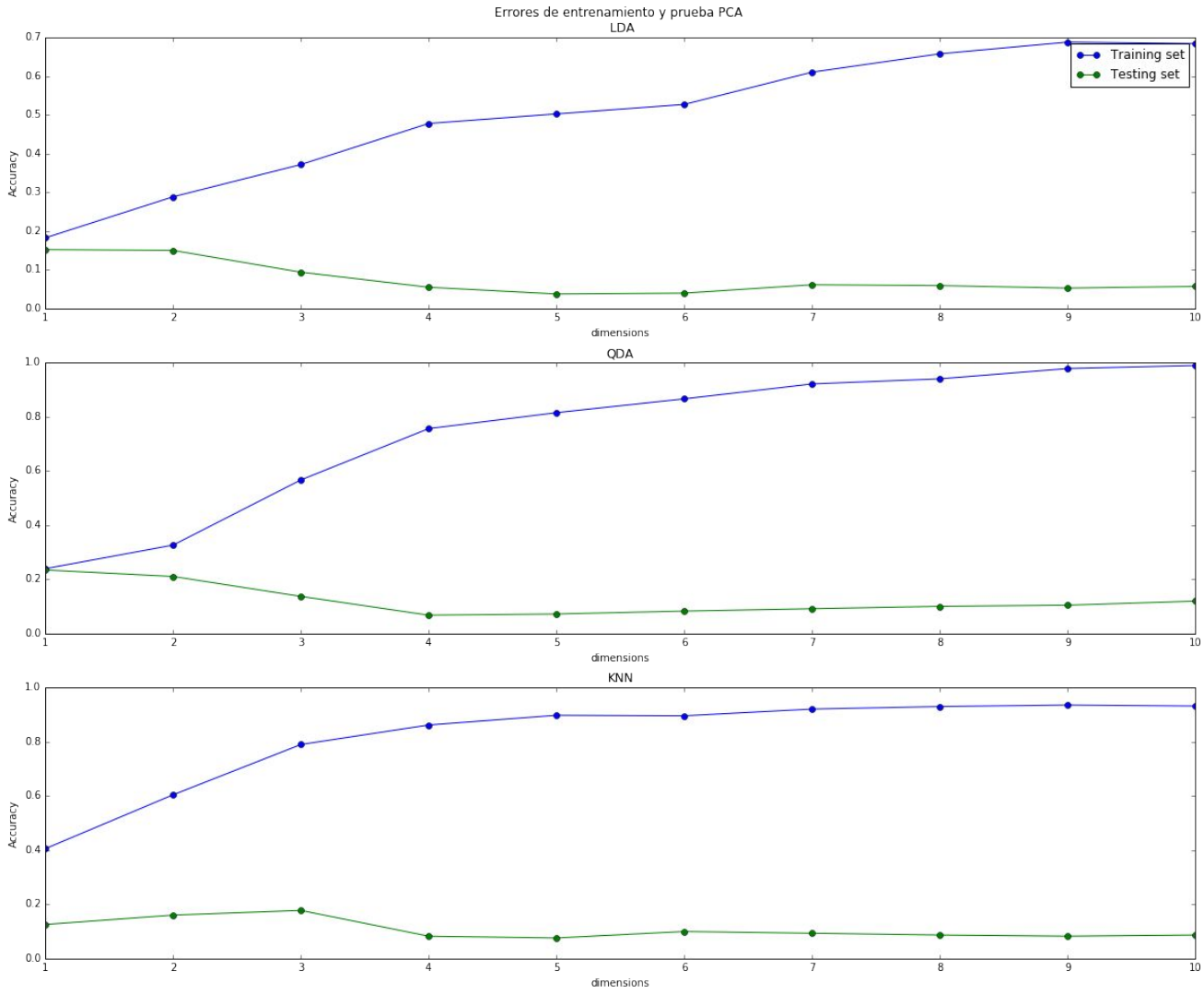
Se observa que para el caso en que K es igual a 1 el modelo tiene un sobreajuste “máximo” a los datos de entrenamiento. Por lo que el modelo no está “aprendiendo”, sino que está “memorizando” los datos de test.

Análisis Cualitativo

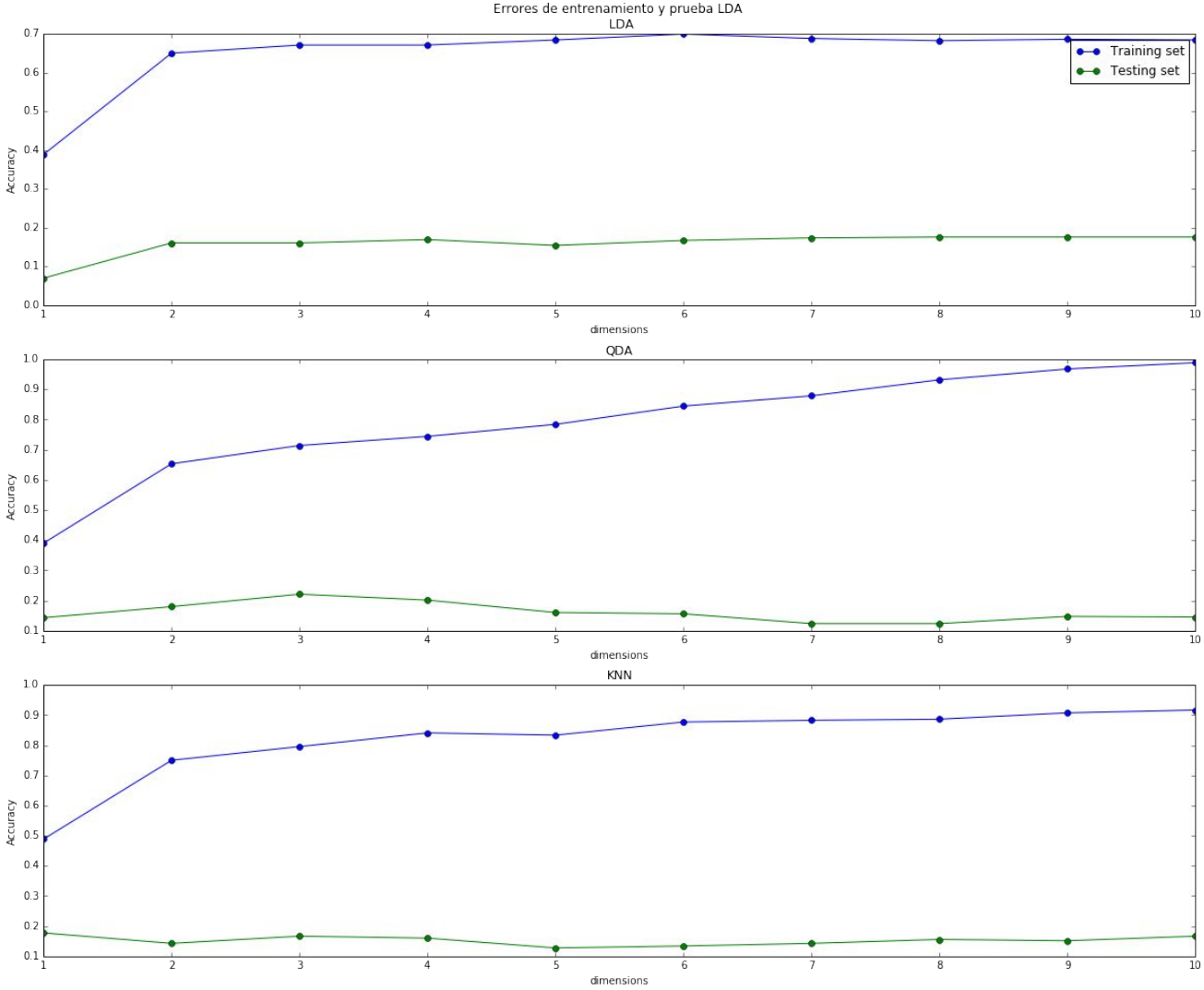
Variación de K para KNN

También se observa que la precisión baja drásticamente a medida que se llega aproximadamente a un K igual a 40 (casi un 8% del dataset de training). Esto puede deberse a que las clases se encuentran traslapadas unas con otras en el espacio de representación 10-dimensional.

Variación de d’ en PCA para distintos modelos



Variación de d' en LDA para distintos modelos



Pregunta 2

Análisis de opiniones sobre películas

Definición del problema.

Se cuenta con un dataset donde cada registro corresponde a una opinión de una películas. El objetivo buscado es predecir si la sentencia es positiva o negativa.

Cantidad de ejemplos pertenecientes a cada clase.

	Clase 1	Clase 0
Train Set	1784	1770
Test Set	1803	1751

Total: 3554

Stemming de Porter

Las técnicas de stemming utilizan un algoritmo para reducir una palabra a su raíz morfológica.
Como es un algoritmo y existen palabras irregulares no funciona en todos los casos.

Ejemplos:

awakened → awaken

resembling → resembl

jumping → jump

communities → commun

Lematización

Busca llevar una palabra a su raíz morfológica. Realiza esta tarea analizando morfológicamente las palabras, por medio de una base de datos léxica.
Se obtienen mejores resultados.

Ejemplos:

awakened → awakened

resembling → resembling

jumping → jump

communities → community

Palabras más frecuentes.

Son 9663 palabras.
Se muestra el Top 10 para Train Set y Test Set.

Frecuencia Train Set	Término Train Set	Frecuencia Test Set	Término Test Set
566	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

Classification Report

Precisión: Proporción de muestras clasificadas como positivas que son positivas.
Precisión = tp / (tp +fp)

Recall: Proporción de muestras positivas que fueron correctamente evaluadas.
recall = tp / (tp + fn)

F1-Score: Media Armónica de la precisión y recall. Relaciona ambas medidas. Varía entre 0 y 1. Siendo 1 lo óptimo.

F1 = 2*(precisión*recall) / (precisión + recall)

Support: Número de ocurrencias de cada clase en vector objetivo.

Resultados Modelos Stemming

	Stemming + StopWords		**Stemming**	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
NB	94.28%	74.78%	93.80%	76.21%
MNB	94.93%	74.97%	94.06%	75.99%
SVM	98.19%	73.12%	98.33%C=0.1	74.05% C=0.1
LR	88.01% C=10	73.12% C=10	100% C=0.1	73.59%

Resultados Modelos Lematización

	Lemmatization + StopWords		**Lemmatization**	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
NB	95.86%	73.85%	95.52%	74.86%
MNB	95.94%	74.07%	95.55%	74.75%
SVM	98.95% C=0.1	72.36% C=0.1	98.79%C=0.1	73.82% C=0.1
LR	89.92% C=10	71.91% C=10	100% C=0.1	73.14%

Textos de ejemplo

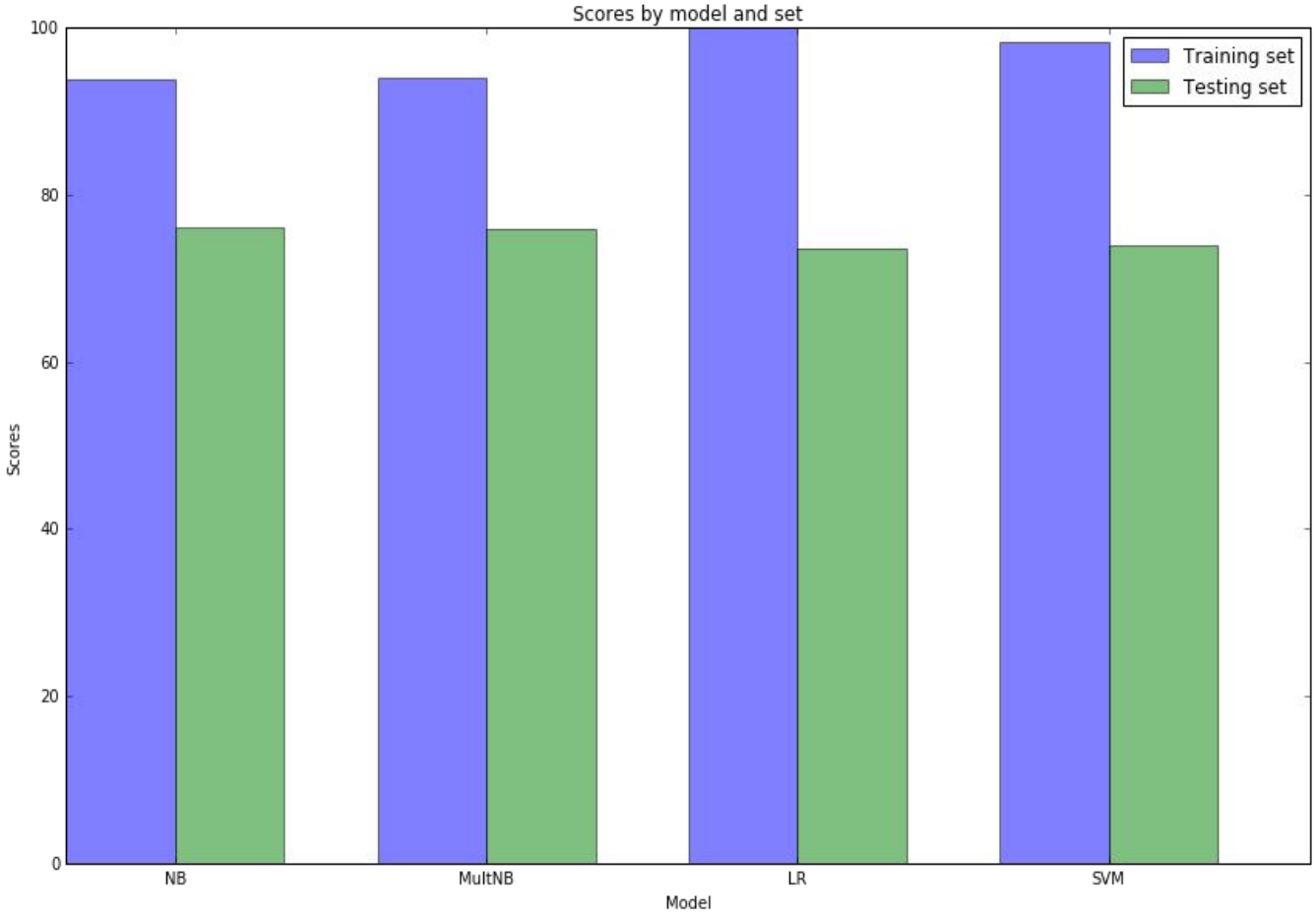
1. the problem is that the movie has no idea of it is serious or not.

NB	MNB	LR	SVM
93.35% - 6.64%	93.27% - 6.72%	72.92% - 27.08%	76.27% - 23.72%

2. wickedly funny , visually engrossing , never boring , this movie challenges us to think about the ways we consume pop culture.

NB	MNB	LR	SVM
7.56% - 92.43%	6.38% - 93.61%	49.33% - 50.66%	43.97% - 56.02%

Precisión vs Modelos



Pregunta 3

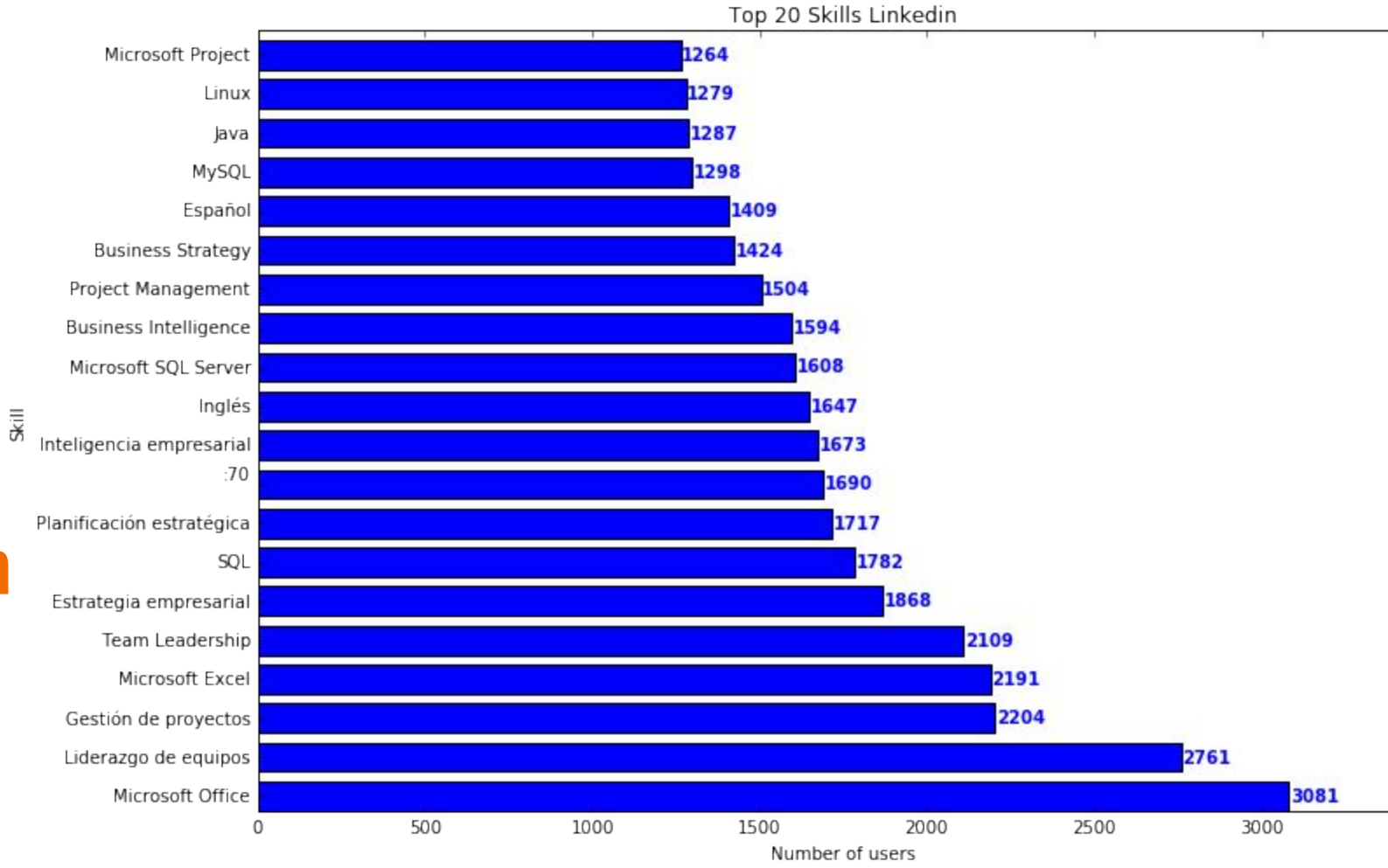
Predicción de Skills de LinkedIn

Definición del problema

Predicción de un skill de interés, que pueda poseer un usuario en base a sus skills declarados en la red.

El dataset cuenta con un total de 7891 ejemplos. Los features de cada ejemplos en total son 14544, que corresponden al total de skills registrados en el archivo skill_id.

TOP 20 skills LinkedIn



Entrenamiento de clasificadores

Se escogieron 4 modelos que dentro de su implementación aceptaban el uso de matrices sparse. Estos fueron: Naive Bayes, Multinomial Naive Bayes, SVM y Logistic Regression.

Se escogieron los skills Microsoft Office, SAP y Business Intelligence para la construcción de los modelos.

Logistic Regression

Para el caso de LR se encontró el parámetro C mediante el método de Grid Search, para valores específicos de C.

Además se utiliza regularización L2 (Ridge) para evitar el sobreajuste del modelo.

Resultados Modelos - Skill Microsoft Office

Skill 29	Train Accuracy	Test Accuracy
NB	84.77%	79.97%
MNB	84.41%	78.91%
SVM 0.01	87.92%	83.10%
LR 0.01	84.06%	83.86%

Resultados Modelos - Skill SAP

Skill 255	Train Accuracy	Test Accuracy
NB	92.77%	89.52%
MNB	93.02%	86.60%
SVM 0.01	94.31%	92.35%
LR 0.1	94.35%	92.18%

Resultados Modelos - Skill Business Intelligence

Skill 8185	Train Accuracy	Test Accuracy
NB	99.98%	100%
MNB	99.49%	98.73%
SVM >0.1	100%	100%
LR 0.01	99.98%	100%