

[Gaussiano + PCA]: Se nos plantea encontrar la mejor combinación de Alphas con PCA para minimizar el error, por ello, empezaremos haciendo un barrido de combinaciones de las distintas alphas que dieron resultados satisfactorios en la implementación de la Gaussiana normal descrita en el anterior ejercicio, que, para unos valores de PCA elegidos nos darán una visión general de como varía el error, así pues, podremos elegir el mejor PCA. Los valores elegidos para las alphas son las que daban los mejores errores en la gaussiana original, así pues, son los siguientes: $\alpha = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1]$ Por otra parte, vamos a probar dichos alphas en los distintos valores de PCA, recordemos que el PCA trabaja con las dimensiones del dataset, por ende, obsérvese que tenemos 784 posibles dimensiones, así pues, para probar más o menos todos los valores, el barrido será a lo largo de estos valores, así que los valores de k para las distintas dimensiones del PCA quedarán como los que siguen:

$k_{PCA} = [1, 5, 10, 15, 20, 25, 30, 31, 32, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 49, 50, 55, 60, 70, 80, 90, 100, 200, 250, 300, 350, 400, 500, 600, 700, 784]$

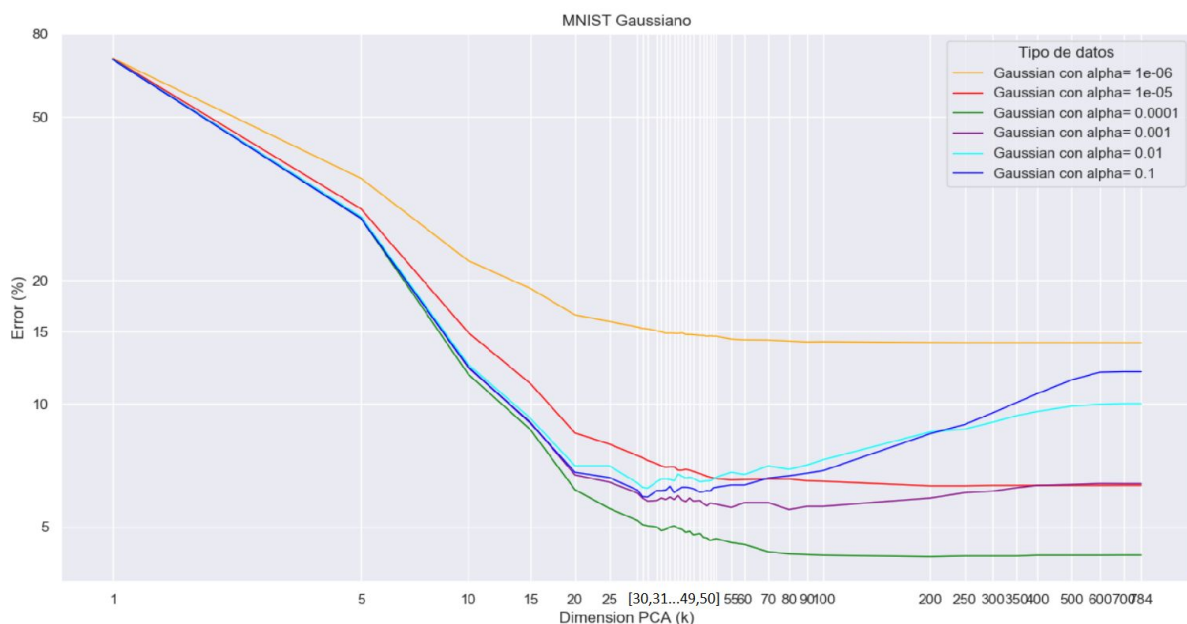


Imagen 1

Como podemos observar en la imagen, una característica en común de las diferentes implementaciones gaussianas es que comienzan con una reducción del porcentaje de error considerable durante la primera lista de dimensiones. No es hasta que llegamos a dimensiones cercanas al $k = 30$ que notamos como algunas comienzan a estabilizarse mientras que otras incluso llegan a su mínimo y comienzan a aumentar sus porcentajes de error, obsérvese las alphas = $[0.1, 0.01, 0.001]$. La alpha verde, 0.001, sin embargo, sigue bajando y por lo tanto será nuestro objetivo de estudio.

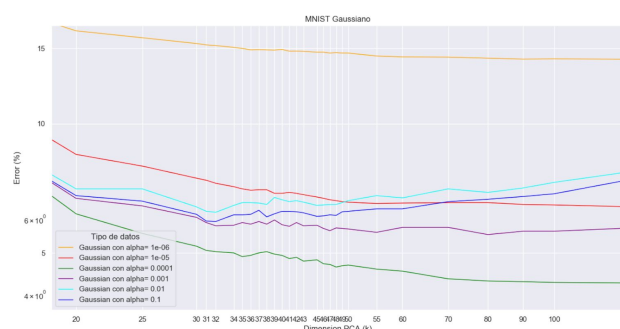


Imagen 2

Tomando una imagen que comprende de las dimensiones 20 a 100 para centrarnos más

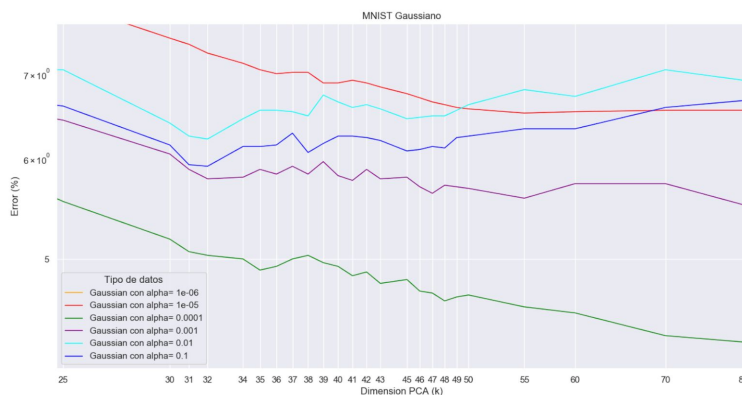


Imagen 3

específicamente en los errores que tenemos, podemos observar esta estabilización: Mientras que las rectas verde, roja y amarilla en mayor o menor medida parece que continúan disminuyendo el porcentaje de error, la azul, cian y morada comienzan a aumentar este porcentaje después de haber tenido bruscas variaciones de rango del $k=40$, lo que nos dice claramente que llegaron a su mínimo, lo que las variaciones del error inestables lo confirman, en la imagen adjunta se puede observar dicha variación de la *Imagen 3*. Por lo tanto, de todas las gaussianas podemos ver que la elegida es la correspondiente al $\alpha = 0.0001$ ya que es la que más ha bajado su porcentaje de error antes de estabilizarse su gráfica, si nos fijamos en su mínimo haciendo una aproximación más específica, nos damos cuenta que el punto mínimo se encontrará cuando llegue a una dimensión $k=200$ (*Imagen 4*)

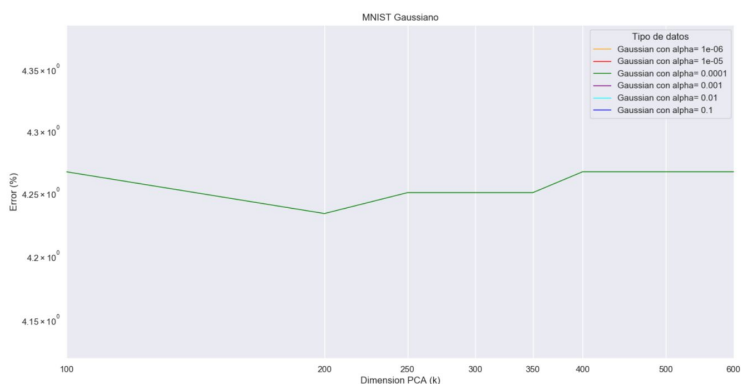


Imagen 4

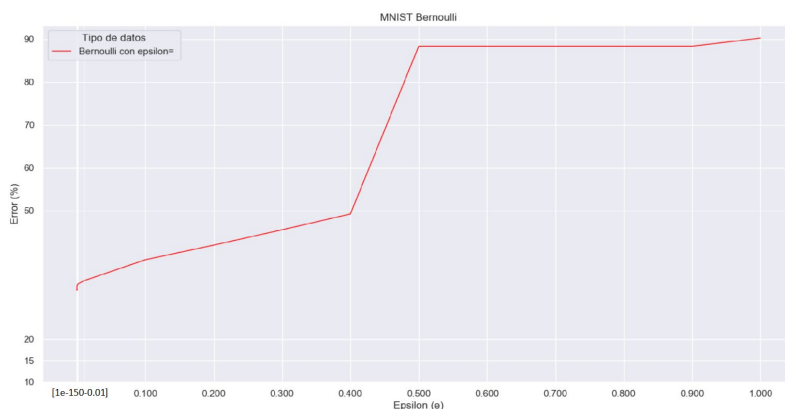
Por último, teniendo la mejor dimensionalidad así como la mejor alpha para ello, llegó la hora de probar el clasificador con dichos parámetros sobre el dataset de testing, así pues, `Alpha= 0.0001` utilizaremos para ello el '*gaussianEVA_PCA.m*', el cual cogerá y calculará el error sobre los datos de testeo usando los parámetros obtenidos en el ejercicio, $\alpha = 0.0001$ y $k = 200$. El resultado que nos sale es de **4.11%**, con lo que mejoramos un 0.07% respecto a la implementación del clasificador gaussiano normal. Cabe destacar que esta mejora no es muy grande, con lo que dependiendo del caso de uso, alomejor nos sería más rentable quedarnos solo con la implementación básica, pero si tuviésemos que elegir un clasificador entre los dos dados, el que se ha hecho utilizando PCA sería el ganador. Recordemos que para los datos dados y la randomización dada, el clasificador nos ha dado esta solución, pero podría dar otra mejor u otra peor también, podríamos intentar hacer otra randomización y buscar una solución mejor.

Imagen 5

Al igual que pasaba con la gaussiana normal, nuestro clasificador es peor del que hay en la web de MNIST (3.3%), pero la diferencia de un 0.8% es muy pequeña, así pues, hemos conseguido implementar un clasificador fiable para ser un clasificador cuadrático y por tanto, interesante para su posterior uso. [Tabla con todos los porcentajes obtenidos en el anexo].

[Bernoulli]: Como uno de los ejercicios opcionales, se nos propone implementar el clasificador de Bernoulli, implementando para ello el experimento de la misma manera que se ha hecho en el multinomial, así pues hemos creado un archivo '*Bernoulli.m*' el cual contiene la implementación del cálculo de Bernoulli. Además, se han implementado funciones auxiliares como '*sigmoid.m*' que, como su nombre indica, aplica el umbral de binarización, ya que tenemos 255 valores para representar las tonalidades de gris del dataset MNIST, por ello, para usar Bernoulli, que es binario, se necesitaba de una binarización. Otra función auxiliar es '*truncarSimplemente.m*', la cual aplica el suavizado por truncamiento simple al dataset, de esta manera nos evitamos tener ceros en la muestra, los cuales son muy peligrosos si observamos la formulación de Bernoulli. Por último, se implementaron los dos archivos para efectuar el experimento, '*BernoulliEXP.m*' y '*BernoulliEVA.m*', el primero nos servirá para obtener la mejor *epsilon*, es decir, la que dará el mínimo error posible sobre los datos de entrenamiento partidos en training set y validation set con proporción 90-10%. La segunda en cambio, aplicará Bernoulli con la mejor epsilon escogida en *BernoulliEXP* para calcular el error sobre el set de testing.

Procediendo con el experimento, hicimos un barrido de epsilons desde $1e-150$ hasta 1.0 , ya que la potencia de cálculo de la máquina lo permitía, obteniendo de resultado la *Imagen 6*.



Como se puede observar, el error empieza a bajar a partir de la *epsilon*= 0.5 , lo que es lógico viendo que es lo que representa epsilon matemáticamente en nuestro clasificador. Obsérvese que de 0.5 a 0.4 hay un salto de error muy grande, ese es nuestro limite donde la epsilon empieza a ser eficiente, a partir de el, vemos que el error empieza a bajar

Imagen 6 - No logarítmica

donde las epsilons empiezan a crecer drásticamente $e = [1e-150, 1e-100, 1e-80 \dots 0.001]$.

Si nos acercamos un poco más a la zona util, podemos observar que a partir de la *epsilon*= 0.1 , el error decrece con más pendiente (*Imagen 7*), así pues a continuación nos

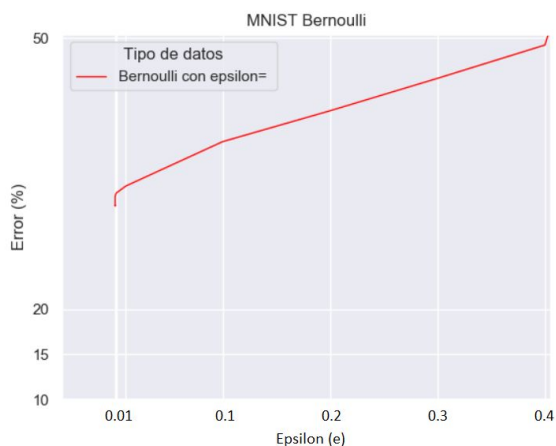


Imagen 7

centraremos en el estudio de la zona que nos llevará introducir epsilons más pequeñas que 0.1 con tal de ver a ver si encontramos alguna donde el error nos puede bajar más aún. Seguidamente nos propusimos hacer un barrido de epsilons siguiente, $e = [1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1]$. En dicha prueba (*Imagen 8*), podemos observar que a partir de la epsilon **$e=0.001$** nuestro error empieza a nivelarse, lo que puede significar que dicha epsilon sería posiblemente una buena solución para el problema, ya que una epsilon más pequeña disminuye el error, aunque ya a pasos muy pequeños. Así pues a modo de una última prueba,

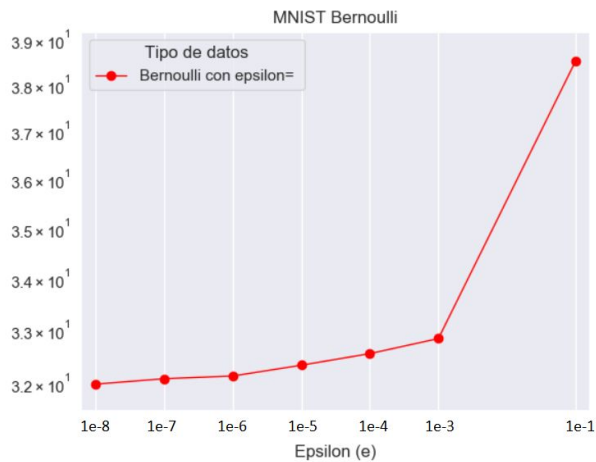


Imagen 8

que tener en cuenta que para cada dataset el error y la epsilon serán distintas, así pues ahora nos interesa mantener el mínimo error en los marcos del experimento. Para un dataset

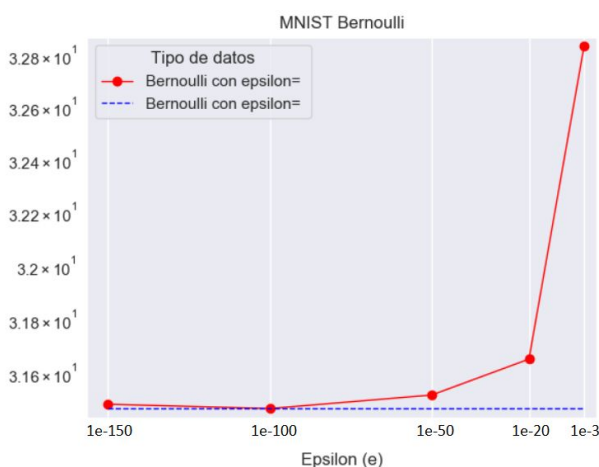


Imagen 9

Por otro lado, si aplicamos $e=1e-20$, nos da un error de **30.07%**, un 0.12% más de error que $e=1e-100$, dependiendo de la tarea en la que usar el clasificador, podríamos usar esta epsilon, pero como ya tenemos un error muy grande, nos quedaremos con la **$e=1e-100$** .

Al terminar, decidimos probar Bernoulli junto a una implementación del PCA vista en la Gaussiana con tal de ver si podemos mejorar el error, ya que 29,95% es muy grande, así pues, hicimos un barrido de dimensionalidades del PCA $k=[1,5,10,20,30,40,50,80,85, 90, 95,100, 110]$, dando la mejor dimensión para nuestra epsilon $e=1e-100$, la **$k=85$** que nos da un error de **26.1%** sobre el set de validación (Imagen 10).

Después de aplicar Bernoulli+PCA sobre datos de test, nos dió un error de **24.72%**, así pues, podemos confirmar que es muy recomendable usar Bernoulli con una reducción de dimensionalidad. Aún así, el clasificador sigue siendo no muy fiable para ciertas tareas peligrosas que requieran mínimo error. [Tablas de errores completas en el anexo].

nos planteamos calcular los errores para las epsilon $e=[1e-150, 1e-100, 1e-50, 1e-20, 1e-3]$, dando un resultado esperado, ya que para $e=1e-100$ el error llega a su mínimo y luego empieza a subir, con lo que ya hemos comprobado que la mejor epsilon sería la **$e=1e-100$** (Imagen 9). Pero si nos fijamos en el error que disminuimos usando una epsilon tan pequeña, este es de solo de **1.34%**, que dentro de lo que cabe, es un error considerable, así pues a la hora de calcular el error para los datos de testeo, usaremos la epsilon **$e=1e-100$** , ya que a pesar de ser un número muy pequeño, tenemos de normal usaríamos una epsilon $e=1e-20$, ya que simplemente perderíamos un 0.15% frente a la epsilon $1e-100$, pero a cambio no operaríamos con números tan pequeños.

Aplicando el '*BernoulliEVA.m*' con **$e=1e-100$** nos da un error de **29.95%**, con lo que hemos bajado algo de error respecto al dataset de training, a pesar de eso, nuestro clasificador no da unos resultados muy buenos, ya que tiene casi un 30% de error, y no sería válido para tareas peligrosas, por ejemplo algo relacionado con la predicción de medicamentos, enfermedades, o cosas que no toleran errores.

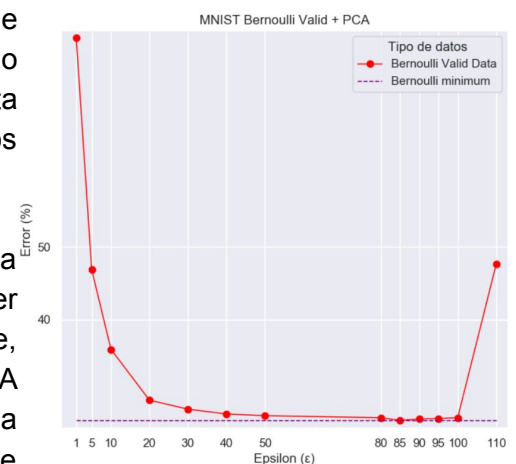


Imagen 10

[Anexo]: Debido al tamaño de las tablas de errores, consideramos que sería mejor dejarlas a parte como dato extra y aprovechar las 2 hojas para centrarnos en la explicación ayudándonos de tablas.

> Gaussiana + PCA:

GAUSSIANO	Errores (%)					
Dimension PCA (k)	1 e-6	1 e-5	1 e4	1 e-3	1 e-2	1 e-1
1	69,67	69,55	69,4	69,38	69,23	69,17
5	35,47	29,88	28,33	28,53	28,6	28,3
10	22,35	14,93	11,82	12,27	12,48	12,32
15	19,1	11,18	8,63	9	9,2	8,95
20	16,47	8,48	6,17	6,7	7,05	6,8
25	15,87	7,97	5,55	6,43	7,05	6,6
30	15,38	7,47	5,18	6,05	6,4	6,15
31	15,27	7,38	5,07	5,88	6,25	5,93
32	15,22	7,27	5,03	5,78	6,22	5,92
34	15,07	7,13	5	5,8	6,45	6,13
35	14,98	7,05	4,9	5,88	6,55	6,13
36	14,87	7	4,93	5,83	6,55	6,15
37	14,88	7,02	5	5,92	6,53	6,28
38	14,87	7,02	5,03	5,83	6,48	6,07
39	14,85	6,88	4,97	5,97	6,73	6,17
40	14,9	6,88	4,93	5,82	6,65	6,25
41	14,77	6,92	4,85	5,77	6,58	6,25
42	14,77	6,88	4,88	5,88	6,62	6,23
43	14,75	6,83	4,78	5,78	6,57	6,2
45	14,68	6,75	4,82	5,8	6,45	6,08
46	14,68	6,7	4,72	5,7	6,47	6,1
47	14,62	6,65	4,7	5,63	6,48	6,13
48	14,65	6,62	4,63	5,72	6,48	6,12
49	14,62	6,58	4,67	5,7	6,55	6,23
50	14,62	6,57	4,68	5,68	6,62	6,25
55	14,38	6,52	4,58	5,58	6,8	6,33
60	14,32	6,53	4,53	5,73	6,72	6,33
70	14,3	6,55	4,35	5,73	7,05	6,58
80	14,22	6,55	4,3	5,52	6,92	6,67
90	14,15	6,48	4,28	5,62	7,08	6,77
100	14,17	6,47	4,27	5,62	7,3	6,87
200	14,1	6,3	4,23	5,88	8,53	8,45
250	14,08	6,3	4,25	6,07	8,67	8,9
300	14,08	6,32	4,25	6,12	9,02	9,5
350	14,08	6,32	4,25	6,23	9,35	10,07
400	14,08	6,32	4,27	6,32	9,57	10,58
500	14,08	6,32	4,27	6,35	9,87	11,43
600	14,08	6,32	4,27	6,38	9,97	11,93
700	14,08	6,32	4,27	6,38	10	11,97
784	14,08	6,32	4,27	6,38	10	11,97

Tabla 1- Errores sobre el Training Set (PCA - Epsilon).

GAUSSIANO EVA + PCA	
Alpha	1 e-4
Testing Set error (%) :	4,11

Tabla 2 - Resultado en testeo

> Bernoulli:

BERNOULLI EXP 1							
Epsilons	1 e-260	1 e-240	1 e-220	1 e-200	1 e-150	1 e-125	1 e-100
Training Set error (%) :	16,09	16,09	16,09	16,09	16,09	16,09	16,09
Validation Set error (%) :	31,5	31,5	31,48	31,48	31,48	31,52	31,47
Epsilons	1 e-80	1 e-50	1 e-20	1 e-6	1 e-5	1 e-4	1 e-3
Training Set error (%) :	16,09	16,09	16,1	16,23	16,26	16,31	16,44
Validation Set error (%) :	31,48	31,52	31,65	32,13	32,33	32,55	32,83
Epsilons	1 e-2	1 e-1	0,2	0,3	0,4	0,5	0,6
Training Set error (%) :	16,88	19,54	22,34	25,28	28,23	88,81	88,81
Validation Set error (%) :	33,62	38,52	41,93	45,52	49,2	88,37	88,37
Epsilons	0,7	0,8	0,9	1			
Training Set error (%) :	88,81	88,81	88,81	90,11			
Validation Set error (%) :	88,37	88,37	88,37				

Tabla 3 - Resultados del experimento 1

BERNOULLI EXP 2								
Epsilons	1 e-8	1 e-7	1 e-6	1 e-5	1 e-4	1 e-3	1 e-2	1 e-1
Training Set error (%) :	16,18	16,19	16,23	16,26	16,31	16,44	16,88	19,54
Validation Set error (%) :	31,98	32,08	32,13	32,33	32,55	32,83	33,62	38,52

Tabla 4 - Resultados del experimento 2

BERNOULLI EXP 3					
Epsilons	1 e-150	1 e-100	1 e-50	1 e-20	1 e-3
Training Set error (%) :	16,09	16,09	16,09	16,1	16,44
Validation Set error (%) :	31,48	31,47	31,52	31,65	32,83

Tabla 5 - Resultados del experimento 3

> Bernoulli + PCA:

BERNOULLI + PCA							
PCA y e=1e-100	1	5	10	20	30	40	
Training Set error (%) :	79,25	54,37	40,15	31,59	28,76	27,61	
Validation Set error (%) :	78,8	46,87	35,88	28,87	27,63	26,97	
PCA y e=1e-100	50	80	85	90	95	100	110
Training Set error (%) :	26,68	25,2	25,06	25,04	24,98	24,95	24,89
Validation Set error (%) :	26,73	26,45	26,1	26,28	26,33	26,47	47,63

Tabla 6 - Resultados del experimento usando PCA

BERNOULLI EVA			BERNOULLI EVA+ PCA	
Epsilons	1 e-100	1 e-20	Epsilons	1 e-100
Testing Set error (%) :	29,95	30,07	Testing Set error (%) :	24,72

Tabla 7 - Resultados finales sobre datos de Testeo

