

# Proyecto final

Álvaro González Redondo, Laura Tirado Lopez

2 de junio de 2016

## Contents

1. Definición del problema a resolver y enfoque elegido	1
2. Codificación de los datos de entrada para hacerlos útiles a los algoritmos.	2
3. Valoración del interés de la variables para el problema y selección de un subconjunto (en su caso).	2
4. Normalización de las variables (en su caso)	2
5. Selección de las técnica (paramétrica) y valoración de la idoneidad de la misma frente a otras alternativas	3
6. Aplicación de la técnica especificando claramente qué algoritmos se usan en la estimación de los parámetros, los hiperparámetros y el error de generalización.	3
7. Argumentar sobre la idoneidad de la función regularización usada (en su caso)	4
8. Valoración de los resultados ( gráficas, métricas de error, análisis de residuos, etc )	4
Clasificación lineal sin decaimiento de pesos: . . . . .	4
Clasificación lineal <i>ridge</i> . . . . .	4
Clasificación lineal <i>lasso</i> . . . . .	5
k-NN . . . . .	6
SVM con kernel de base radial . . . . .	7
Red neuronal multicapa . . . . .	8
9. Justificar que se ha obtenido la mejor de las posibles soluciones con la técnica elegida y la muestra dada. Argumentar en términos de la dimensión VC del modelo, el error de generalización y las curvas de aprendizaje.	9

## 1. Definición del problema a resolver y enfoque elegido

El problema que queremos abordar es el **reconocimiento de la actividad humana basado en datos del teléfono móvil**.

El **Reconocimiento de la Actividad Humana (HAR)** en sus siglas en inglés) trata de identificar las acciones llevadas a cabo por una persona dado un conjunto de observaciones de la misma y de su contexto<sup>1</sup>.

---

<sup>1</sup>[Anguita et al.] *A Public Domain Dataset for Human Activity Recognition Using Smartphones*. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz.

Los **teléfonos móviles inteligentes (smartphones)** son dispositivos que la mayoría de la gente lleva de forma continuada con comodidad, y disponen de suficientes sensores (acelerómetros, micrófonos...) y capacidad de cómputo. Por ello los smartphones son apropiados para realizar HAR.

El artículo de [Anguita et al.] describe como identificar 6 tipos de acciones mediante un clasificador SVM multiclase a partir de datos obtenidos de smartphones. Como proyecto final nos gustaría intentar replicar sus resultados con el mismo clasificador y base de datos.

El **enfoque** que hemos elegido es aplicar cada uno de los modelos que hemos ido viendo en prácticas en orden de complejidad creciente:

- Clasificación lineal, sin y con decaimiento de pesos (*lasso* y *ridge*).
- *k*-Nearest Neighborhood.
- Support Vector Machines, con kernel de base radial.
- Red neuronal multicapa.

Para aquellos modelos en los que es necesario elegir parámetros se ha optado por hacer un **grid-search** de los mismos y elegir aquellos con menor error de validación cruzada.

## 2. Codificación de los datos de entrada para hacerlos útiles a los algoritmos.

La base de datos consta de 561 variables de entrada, muy preprocesadas.

Las primeras variables, tomadas por el acelerómetro y el giroscopio, son la aceleración lineal y la velocidad angular. Se toman series de datos en ventanas temporales de 2.56 segundos a 50 Hz. Estas señales se preprocesaron para reducir el ruido de los datos. La señal de aceleración fue dividida en dos componentes, en la aceleración del cuerpo y gravedad. Además se obtuvieron otras señales adicionales a partir de las recogidas inicialmente por los dispositivos. En total, se obtienen 17 señales a las cuáles se le aplican distintos cálculos como el ángulo entre los vectores, media, correlación, desviación típica, máximo valor, mínimo valor, etc. Tras aplicar esta serie de cálculos a las 17 señales se obtienen las 561 variables.

Aporta un conjunto de training con 7352 casos y otro de test con 2947 casos. Las personas (30 en total) que han aportado los datos para cada conjunto son diferentes.

## 3. Valoración del interés de la variables para el problema y selección de un subconjunto (en su caso).

El conjunto de variables obtenido por mapeado de características contiene resumida la información de las señales tomadas por los sensores. Los autores originales no hicieron selección de un subconjunto de variables para entrenar el SVM, y nosotros no hemos visto necesario hacerlo con ninguno de los otros modelos.

## 4. Normalización de las variables (en su caso)

Los autores de la base de datos normalizaron los datos entre -1 y 1.

## 5. Selección de la técnica (paramétrica) y valoración de la idoneidad de la misma frente a otras alternativas

Las técnicas que hemos utilizado han sido:

- Clasificación lineal sin decaimiento de pesos.
- Clasificación lineal ridge.
- Clasificación lineal lasso.
- k-NN.
- SVM con kernel de base radial, multiclase con One-Vs-All (OVA).
- Red neuronal multicapa.

Los autores de la base de datos entrenaron un modelo SVM para probar que era útil y hemos querido replicar sus resultados usando dicho modelo.

Dado que SVM es una técnica con cierta complejidad, decidimos empezar probando modelos más sencillos antes de implementar SVM para comprobar si técnicas con mayor capacidad de generalización no daban mejores resultados para esta base de datos. A su vez, también quisimos probar con una técnica más sofisticada como lo son las redes neuronales para ver si ofrece mejores resultados que SVM.

## 6. Aplicación de la técnica especificando claramente qué algoritmos se usan en la estimación de los parámetros, los hiperparámetros y el error de generalización.

Para aplicar las técnicas hemos estimado los hiperparámetros, luego hemos entrenado un modelo para cada técnica con esos hiperparámetros, y finalmente hemos calculado su error de generalización.

Para **estimar los hiperparámetros** de los modelos hemos implementado búsqueda grid para cada una de las técnicas para buscar los hiperparametros más idóneos. El espacio de búsqueda del algoritmo es el producto cartesiano de los conjuntos de los distintos valores de las variables. Para cada modelo, los valores de las variables utilizadas han sido:

- Clasificación lineal sin decaimiento de pesos: no se ha realizado búsqueda grid.
- Clasificación lineal ridge:  $\lambda \in \{10^{-3}, \dots, 10^6\}$ .
- Clasificación lineal lasso:  $\lambda \in \{10^{-10}, \dots, 10^{-1}\}$ .
- k-NN:  $k \in \{2, 3, 4, 8, 16, 32\}$ .
- SVM:
  - $C \in \{10^{-1}, \dots, 10^4\}$
  - $\gamma \in \{10^{-4}, \dots, 1\}$
- Red neuronal:
  - N° de capas y neuronas:  $[64, 32], [32, 10], [10, 5], [64], [32], [10]$
  - Regularización l2:  $[1, 0.1, 0.01, 0.001, 0.0001, 0]$
  - Método de optimización:  $[SGD, adadelta]$

Para la **estimación del error de generalización** utilizamos validación cruzada de 10 particiones en todas las técnicas implementadas.

La **estimación de los parámetros** es calculada por cada una de las técnicas que hemos implementado, para las cuáles se han escogido las hipótesis que tuviesen menor error de generalización gracias a la búsqueda grid y a la validación cruzada.

## 7. Argumentar sobre la idoneidad de la función regularización usada (en su caso)

Para el SVM se ha regularizado mediante el parámetro  $C$  (coste).

Para la red neuronal, como hemos ajustado un modelo de regularización *ridge* y otro de regularización *lasso*, aplicamos *lasso* como función de regularización al dar mejores resultados que *ridge*.

## 8. Valoración de los resultados ( gráficas, métricas de error, análisis de residuos, etc )

Los resultados para cada una de las técnicas aplicadas son los siguientes:

### Clasificación lineal sin decaimiento de pesos:

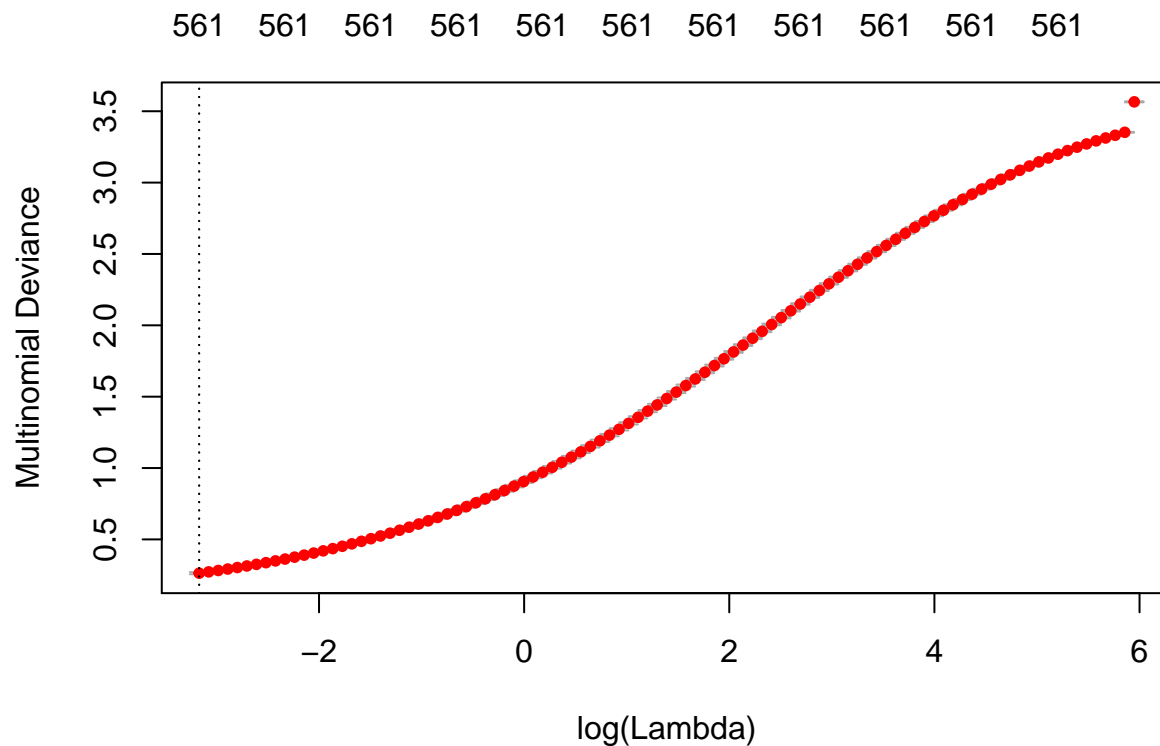
Los resultados del modelo son:

```
## [1] "Error de clasificación en el conjunto test del modelo lineal: 0.0536138445877163"
```

```
##          Y_test
## y      LAYING SITTING STANDING WALKING WALKING_DOWNSTAIRS
## LAYING      536      8        0        1          0
## SITTING      0     421     20        1          0
## STANDING      0     56    510        0          8
## WALKING       1      1      1     492          5
## WALKING_DOWNSTAIRS 0      0      1      0         400
## WALKING_UPSTAIRS  0      5      0      2          7
##          Y_test
## y      WALKING_UPSTAIRS
## LAYING                0
## SITTING                0
## STANDING               3
## WALKING               37
## WALKING_DOWNSTAIRS     1
## WALKING_UPSTAIRS      430
```

### Clasificación lineal *ridge*

Los resultados de grid-search son:



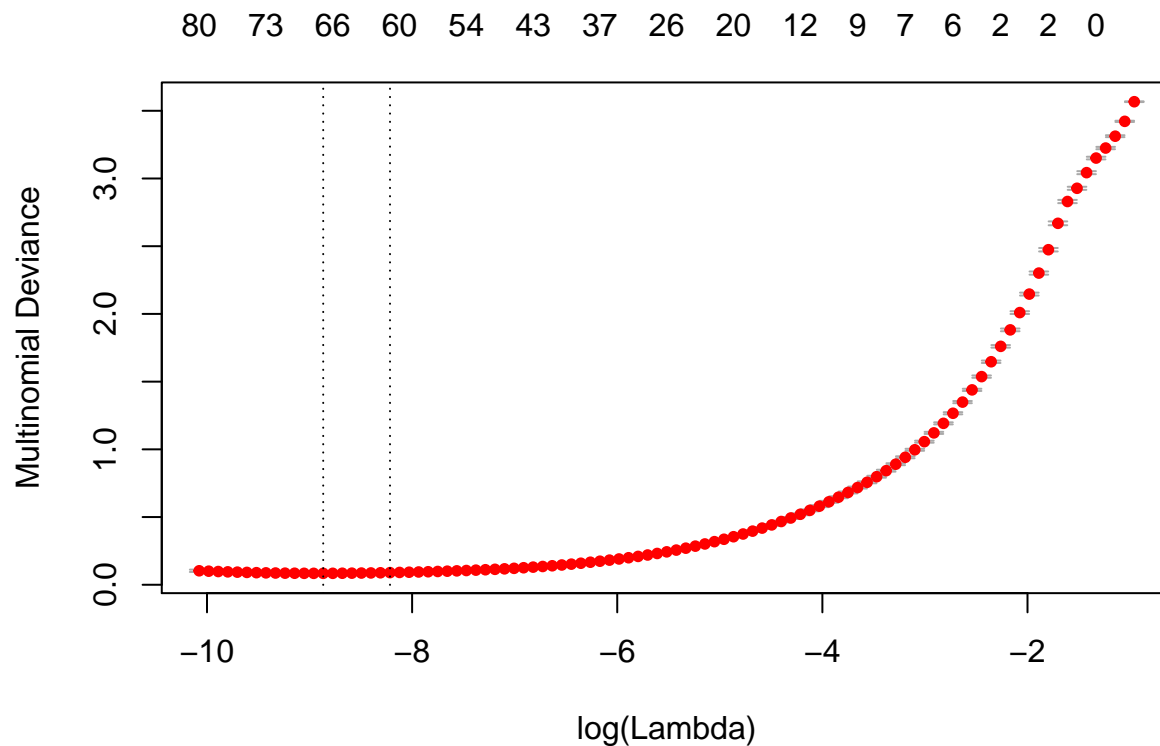
El valor del parámetro  $\lambda$  elegido es 0.04206891. Los resultados del modelo son:

```
## [1] "Error de clasificación en el conjunto test del modelo lineal con weight decay (ridge): 0.058025110281"
```

```
##          Y_test
## y      LAYING SITTING STANDING WALKING WALKING_DOWNSTAIRS
## LAYING      524      0        0        0                0
## SITTING      0     434      31        0                0
## STANDING     13      54     501        0                0
## WALKING       0      0        0     492               14
## WALKING_DOWNSTAIRS 0      0        0      4             380
## WALKING_UPSTAIRS  0      3        0      0              26
##          Y_test
## y      WALKING_UPSTAIRS
## LAYING                0
## SITTING                0
## STANDING               1
## WALKING               23
## WALKING_DOWNSTAIRS     2
## WALKING_UPSTAIRS     445
```

## Clasificación lineal *lasso*

Los resultados de grid-search son:



El valor del parámetro  $\lambda$  elegido es 0.0001409983. Los resultados del modelo son

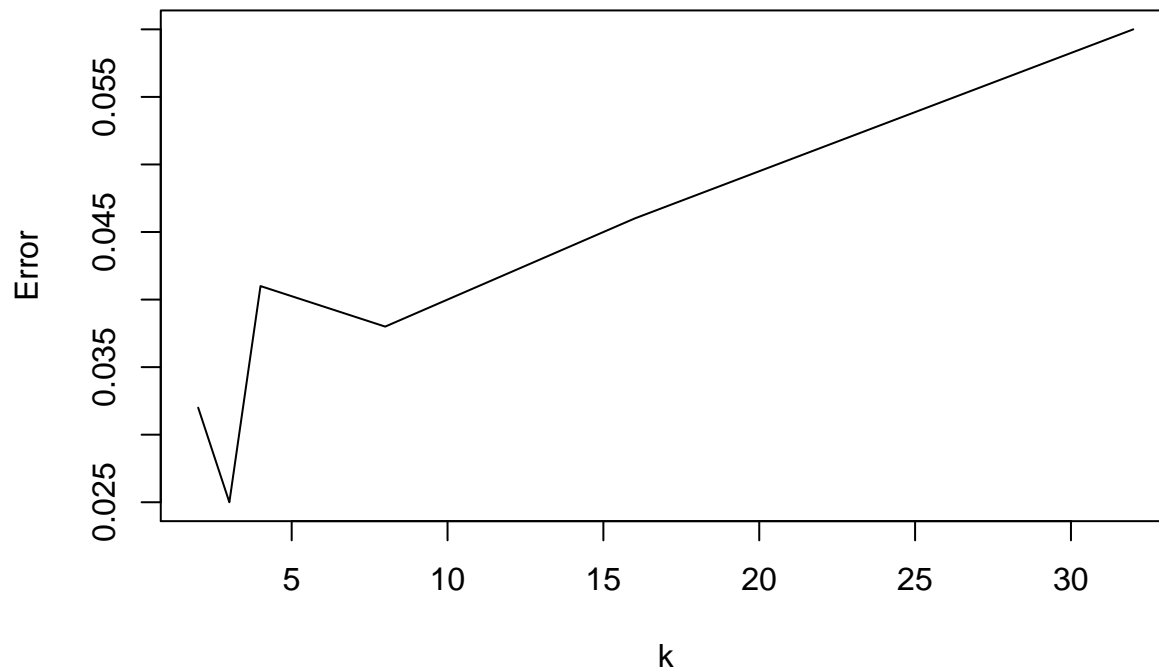
```
## [1] "Error de clasificación en el conjunto test del modelo lineal con weight decay (lasso): 0.048184594502"
```

```
##
##          Y_test
## y      LAYING SITTING STANDING WALKING WALKING_DOWNSTAIRS
## LAYING      535      0      0      0      0
## SITTING      0     426     14      0      0
## STANDING      2     63    517      0      1
## WALKING      0      0      1    496      2
## WALKING_DOWNSTAIRS 0      0      0      0    396
## WALKING_UPSTAIRS  0      2      0      0     21
##
##          Y_test
## y      WALKING_UPSTAIRS
## LAYING                0
## SITTING                1
## STANDING               0
## WALKING               33
## WALKING_DOWNSTAIRS     2
## WALKING_UPSTAIRS     435
```

## k-NN

Los resultados de grid-search son:

## Error de clasificación en función del hiperparámetro $k$



El mejor valor de  $k$  ha sido 3. Los resultados del modelo entrenado con ese parámetro son:

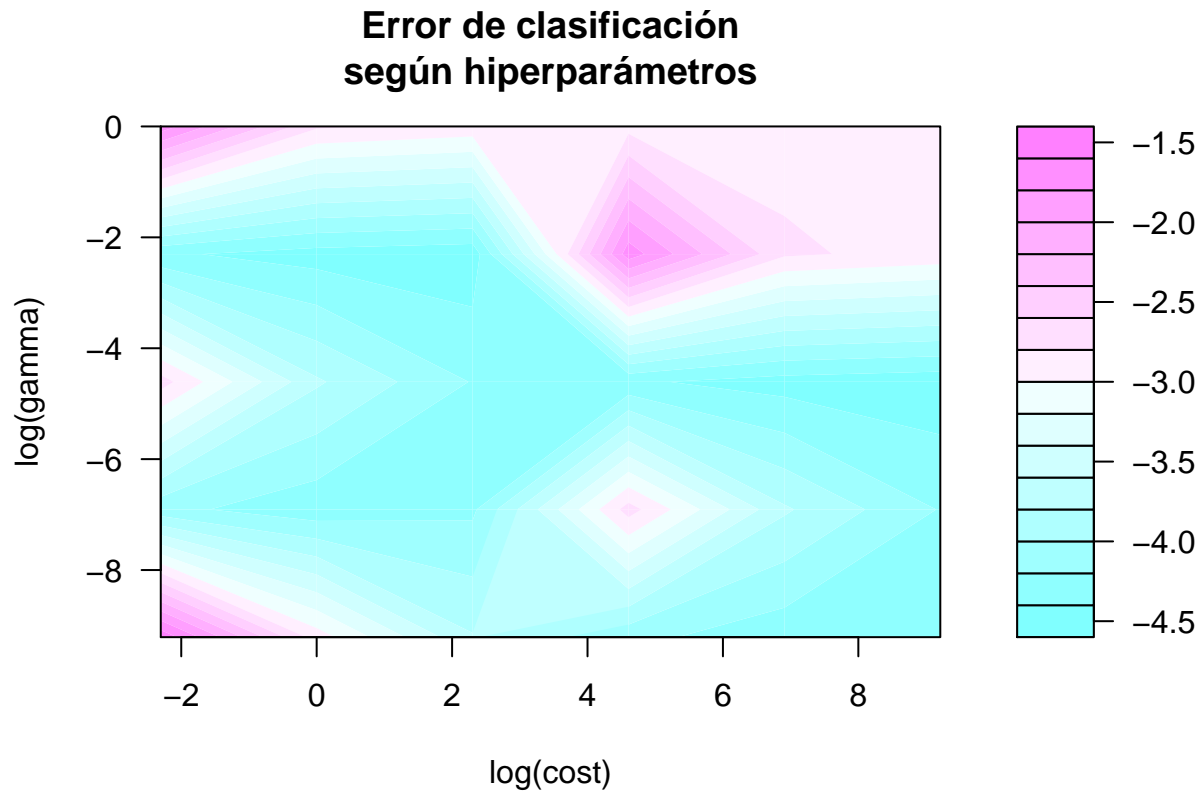
```
## [1] "Error de clasificación en el conjunto test del 3-NN: 0.109263657957245"
```

```
##          Y_test
## y      LAYING SITTING STANDING WALKING WALKING_DOWNSTAIRS
## LAYING      535      0        0        0                0
## SITTING      1     383      57        0                0
## STANDING      1     105     475        0                0
## WALKING       0      0        0     479                47
## WALKING_DOWNSTAIRS 0      0        0      12             329
## WALKING_UPSTAIRS  0      3        0       5              44

##          Y_test
## y      WALKING_UPSTAIRS
## LAYING                  0
## SITTING                 0
## STANDING                 0
## WALKING                  31
## WALKING_DOWNSTAIRS       16
## WALKING_UPSTAIRS       424
```

## SVM con kernel de base radial

Los resultados de grid-search son:



Los hiperparámetros con mejores resultados han sido  $C = 10^4$  y  $\gamma = 10^{-3}$ . Los resultados del modelo entrenado con esos parámetros son:

```
## [1] "Error de clasificación en el conjunto test del SVM: 0.0410587037665422"
```

```
##
## y
## LAYING SITTING STANDING WALKING WALKING_DOWNSTAIRS
## LAYING 537 2 0 0 0
## SITTING 0 439 11 0 0
## STANDING 0 49 521 0 0
## WALKING 0 0 0 486 6
## WALKING_DOWNSTAIRS 0 0 0 4 389
## WALKING_UPSTAIRS 0 1 0 6 25
##
## y
## WALKING_UPSTAIRS
## LAYING 0
## SITTING 0
## STANDING 0
## WALKING 15
## WALKING_DOWNSTAIRS 2
## WALKING_UPSTAIRS 454
```

## Red neuronal multicapa

En la búsqueda grid el mejor modelo ha sido con método de optimización *adadelata*, factor de regularización *lasso* 0.0001, y dos capas con 64 y 32 neuronas. El error de test es  $E_{test} = 0.0617$ , y la matriz de confusión es



##	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
## [1,]	526	0	0	0	0	0
## [2,]	0	393	14	0	0	0
## [3,]	11	96	518	0	0	1
## [4,]	0	0	0	482	4	14
## [5,]	0	0	0	9	392	2
## [6,]	0	2	0	5	24	454

**9. Justificar que se ha obtenido la mejor de las posibles soluciones con la técnica elegida y la muestra dada. Argumentar en términos de la dimensión VC del modelo, el error de generalización y las curvas de aprendizaje.**

Podemos utilizar el conjunto de test como conjunto de validación,  $\mathcal{D}_{val}$ . Tomamos el conjunto de los modelos de cada técnica como un conjunto de hipótesis de validación,  $\mathcal{H}_{val} = \{g_{lineal}, g_{lin\_ridge}, \dots, g_{red\_neural}\}$ . Como este conjunto se ha obtenido sin información sobre el conjunto de validación, elegir una hipótesis  $g_*$  en función del error de clasificación  $E_{val}(g_*)$  en el conjunto de validación  $\mathcal{D}_{val}$  es equivalente a aprender una hipótesis  $g_*$  del conjunto de hipótesis  $\mathcal{H}_{val}$  usando los datos del conjunto de validación  $\mathcal{D}_{val}$ . Aquí los errores de cada hipótesis  $g_*$  son errores dentro de la muestra y podemos aplicar la cota VC para conjuntos de hipótesis finitos. La expresión sería:

$$E_{out}(g_*) \leq E_{val}(g_*) + \mathcal{O}\left(\sqrt{\frac{\log |\mathcal{H}_{val}|}{|\mathcal{D}_{val}|}}\right) = E_{val}(g_*) + \mathcal{O}\left(\sqrt{\frac{\log 6}{2947}}\right)$$

Lo que nos da una cota muy buena.

Por tanto podemos decir que la técnica que da mejores resultados es aquella que tiene el menor error de validación (que tiene la cota de error fuera de la muestra más baja). Las cotas de los modelos son las siguientes:

- Clasificación lineal sin decaimiento de pesos: 0.07827139
- Clasificación lineal ridge: 0.08268265
- Clasificación lineal lasso: 0.07284214
- k-NN: 0.1339212
- SVM con kernel de base radial: 0.06571624
- Red neuronal: 0.08641526

Por tanto, el mejor modelo es el SVM con kernel de base radial.