WINE QUALITY

PROYECTO DE APRENDIZAJE AUTOMÁTICO

LIN M. DOTOR

Índice

Introducción

Esta práctica se basa en emplear los distintos métodos aprendidos en la asignatura en un dataset de nuestra elección, para desarrollar un sistema de aprendizaje automático. Eligiendo un problema de clasificación podremos emplear más técnicas.

También se ofrecerá una estimación de la efectividad del sistema.

Además, describiremos qué técnicas se han utilizado y por qué, valorando los resultados obtenidos y añadiendo gráficas para explicar la solución.

El DataSet que he utilizado es el de “Wine Quality”, que se puede obtener en las web de UC Irvine Machine Learning Repository

Realizaremos 2 tipos de aprendizaje:

- Adivinar la nota que se le da a un vino (tinto o blanco) en función de sus propiedades químicas.
- Diferenciar entre un vino tinto y un vino blanco.

Como en clase hemos aprendido muchos métodos de Aprendizaje Automático (regresión lineal, regresión logística, redes neuronales, support vector machines…) vamos a emplear 3 de estos métodos:

- Regresión Lineal con varios atributos
- Regresión Logística
- Regresión Logística Multiclase
- Redes Neuronales.

Preparación de los datos

En la primera parte vamos a tomar el dataset (datos de vino blanco y vino tinto) como dataset independientes, intentando aplicar en ello distintas técnicas para "adivinar" la nota que se le daría a cada vino, en función de sus propiedades.

Los atributos del DataSet son los siguientes:

VARIABLES DE ENTRADA:
1 - fixed acidity
2 - volatile acidity
3 - citric acid
4 - residual sugar
5 - chlorides
6 - free sulfur dioxide
7 - total sulfur dioxide
8 - density
9 - pH
10 - sulphates
11 - alcohol
VARIABLE DE SALIDA:
12 - quality (score between 0 and 10)

Lo primero que tenemos que hacer es preparar y cargar los datos, para ello hemos tomado los ficheros .csv y hemos eliminado las cabeceras de texto que definen qué significa cada columna. Nos basta con saber que las 11 primeras columnas son propiedades, y las última es la nota que se le da a cada vino. Además hemos sustituido el delimitador ";" por el de ",", que entiende por defecto la función load de Octave.

Para comenzar a hacer pruebas hemos reducido el número de ejemplos de entrenamiento, y nos hemos quedado con 50 de los 5000 posibles, para poder hacer pruebas sobre un conjunto de datos más "manejable".

Una vez que comprobamos que los algoritmos funcionan, ampliamos las muestras al total de los ejemplos de entrenamiento, dejando al margen un 10% de ejemplos (siempre aleatorios) para realizar las pruebas de validación con ejemplos que no se hayan utilizado en el entrenamiento.

Un dato a destacar acerca del DataSet es que no está balanceado, esto es que hay muchas muestras con valores intermedios (5,6, 7) y muy pocas muestras con valores muy elevados o muy bajos. Los porcentajes de cada una de las notas son los siguientes:

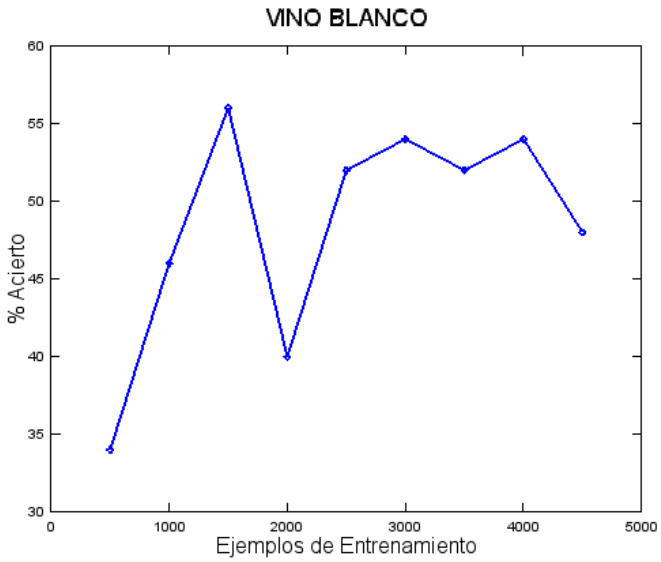
NOTA	Nº EJEMPLOS	PORCENTAJE
0	0	0,00%
1	0	0,00%
2	0	0,00%
3	30	0,46%
4	216	3,33%
5	2138	32,93%
6	2831	43,61%
7	1079	16,62%
8	193	2,97%
9	5	0,08%
10	0	0,00%
TOTAL	6492	100,00%

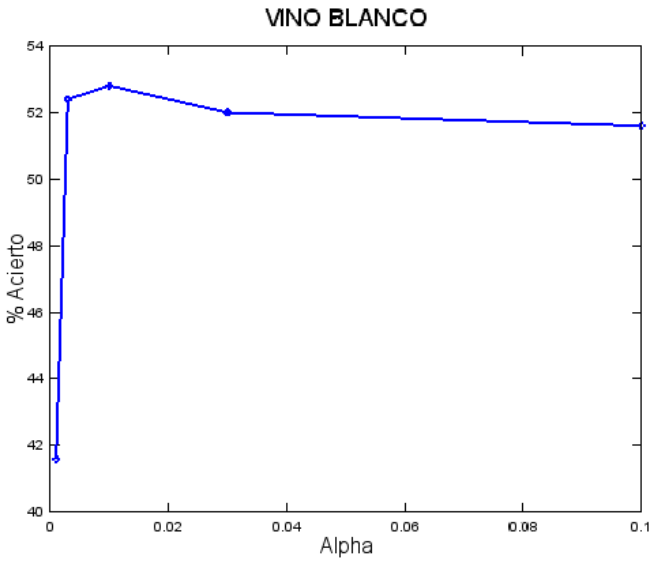
Regresión lineal

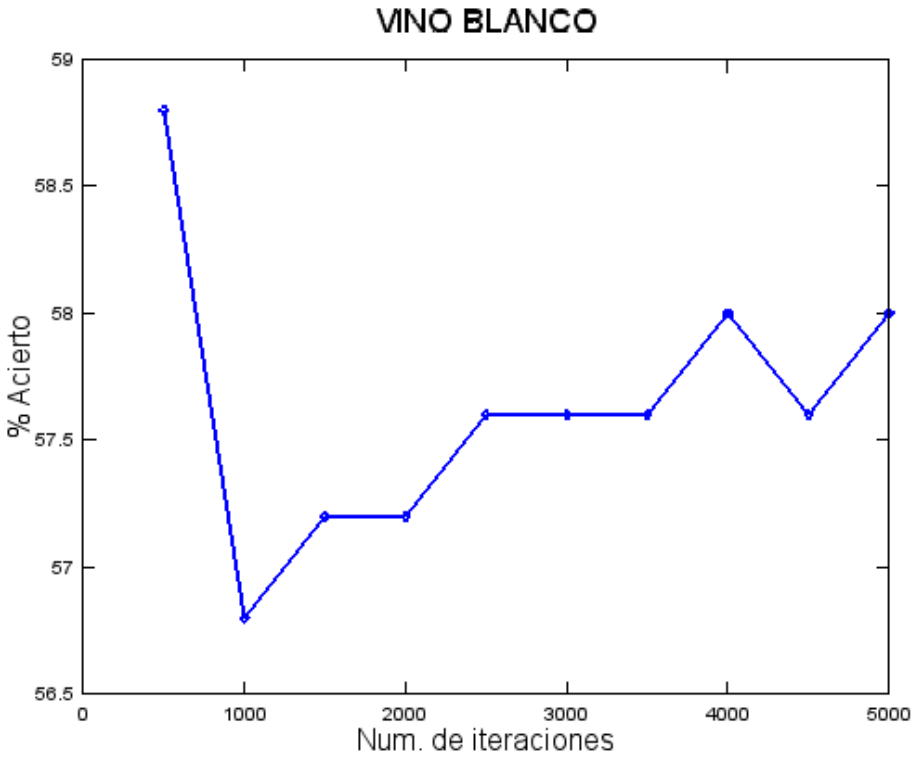
Hemos aplicado la regresión lineal con varias variables sobre ambos conjuntos de datos (vino tinto y vino blanco), y hemos hecho la prueba de clasificación, para ver si le resulta fácil adivinar entre los 2 tipos de vino, entre muestras que no se han utilizado para la clasificación.

Vino Blanco

Si nos fijamos en las gráficas de acierto para el problema de adivinar la nota en los vinos de tipo blanco, en función del número de ejemplos de entrenamiento, del parámetro landa, y el número de iteraciones, podemos obtener las siguientes gráficas:





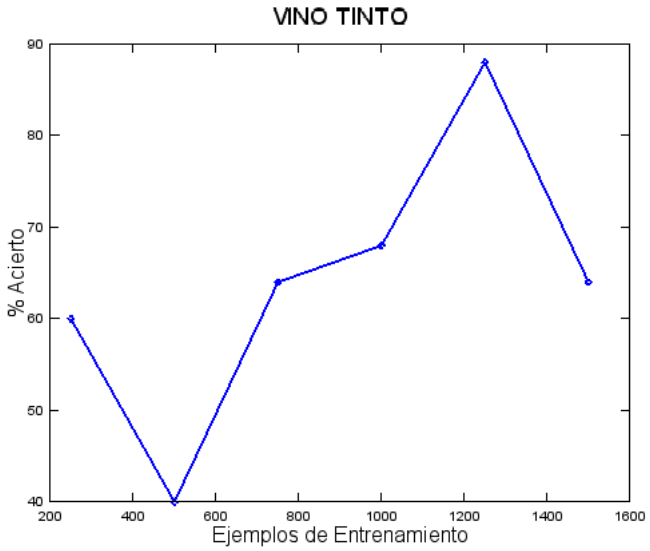


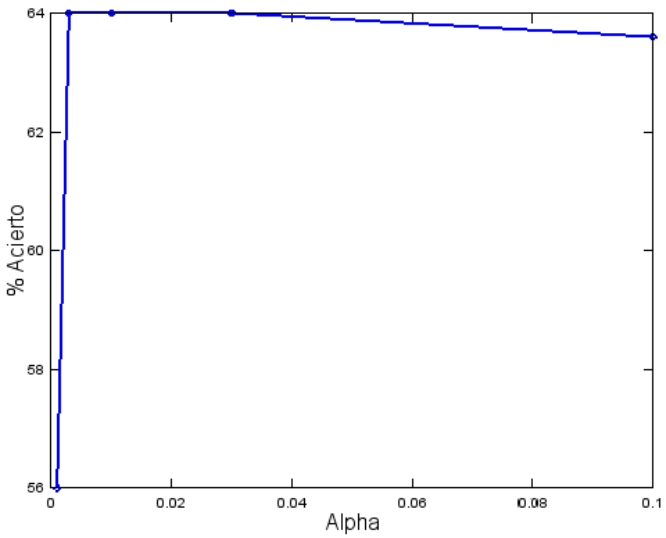
En el vino blanco podemos observar que (como era de esperar) según crece el número de iteraciones aumenta la precisión. A pesar de ello el número de ejemplos de entrenamiento no es un parámetro que influya demasiado. Para el valor alpha, obtenemos un máximo en torno a 0.01, y luego se estabiliza.

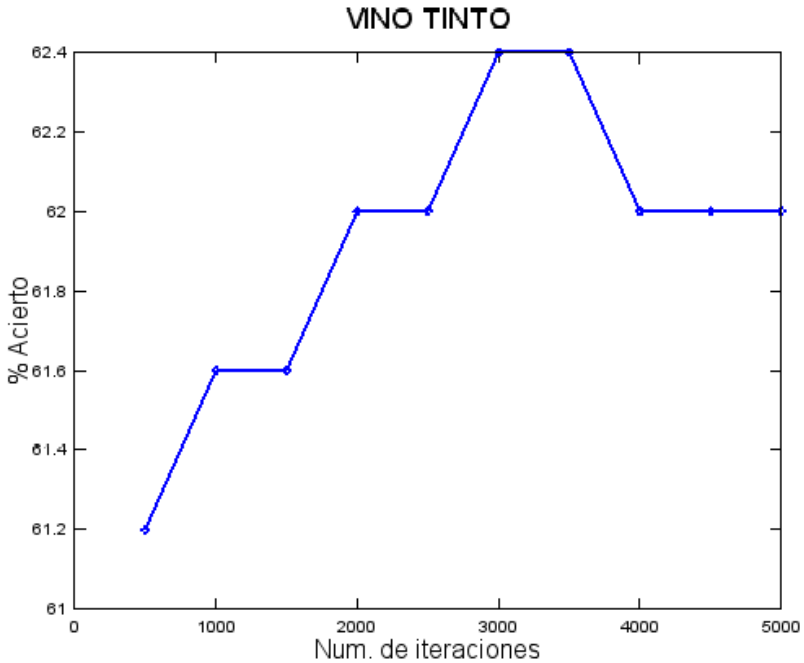
Para este problema, los valores de precisión máximos que hallamos se encuentran en torno al 55%-58%, que no son muy elevados.

Vino Tinto

Para el vino tinto obtenemos las siguientes:



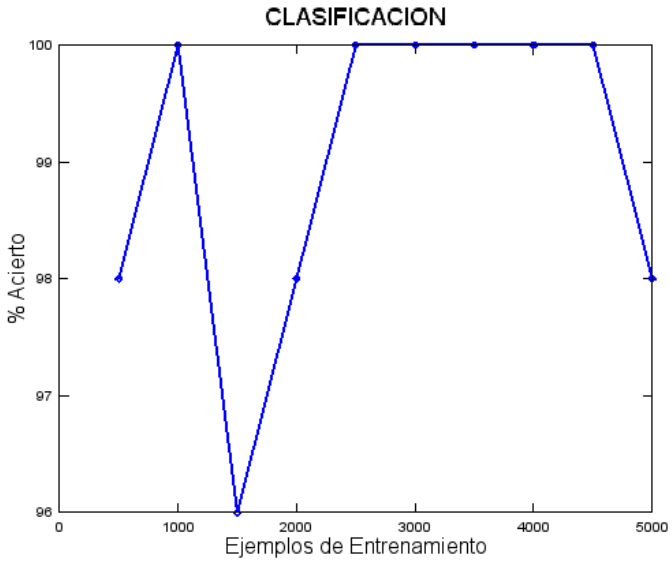


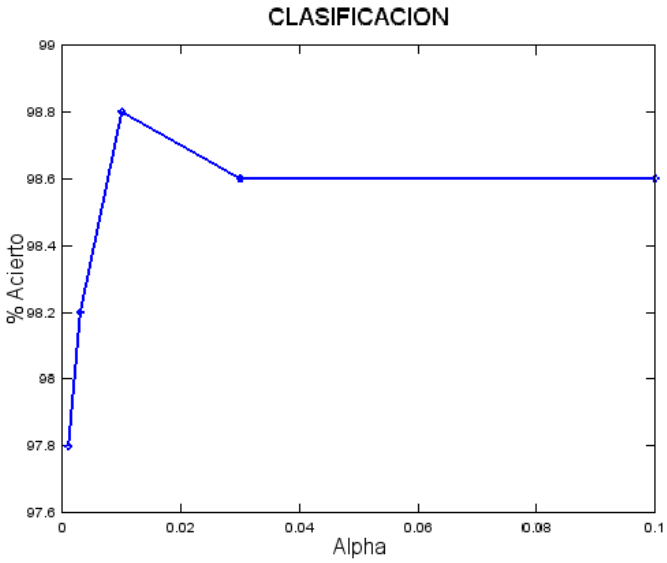


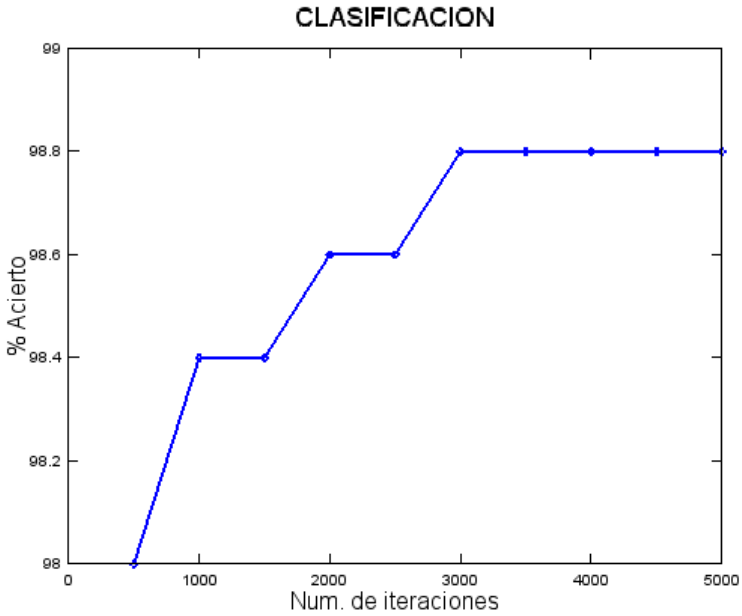
Podemos observar que los comportamientos generales de las gráficas son similares, aunque valores de precisión para este problema son algo mayores que para el del vino blanco, obteniendo valores entre 60%-64% en la mayor parte de los casos, y algunos máximos de 80% según aumentan el número de ejemplos de entrenamiento.

Diferenciar tipos de vino

Para el problema de clasificar los ejemplos de entrenamiento (tinto o blanco) correctamente, hemos obtenido las siguientes gráficas:







En estos ejemplos podemos observar que el número de iteraciones afecta de manera muy clara al acierto, que el valor alpha sigue el mismo comportamiento que antes, y que el número de ejemplos de entrenamiento no es muy “aclarador”. A pesar de ello, el porcentaje de acirto en el problema de clasificación, es bastante más elevado, situándose en el mejor caso en torno al 98%-99%

Como conclusión de la regresión lineal podemos decir que la regresión lineal no es el mejor método para este tipo de datos, para adivinar la nota, porque al generar una función lineal, la predicción posterior al aprendizaje es bastante limitada. Aunque sí se acerca bastante a los resultados, no termina de funcionar bien para muchos ejemplos de entrenamiento.

En cambio para el proceso de distinguir entre los 2 tipos de vino, se aplica con un porcentaje muy alto de precisión (0.98%) por lo que la regresión lineal para el problema de clasificación funciona bastante bien.

Regresión logística

En esta parte vamos a intentar aplicar la regresión logística sólo para el problema de adivinar el tipo de vino por los atributos (2 clases).

Dado a que tenemos 11 atributos y miles de ejemplos de entrenamiento, vamos a aplicar directamente la regresión logística regularizada, ya que no podríamos visualizar a primera vista si los datos se pudieran separar claramente con una función lineal o no.

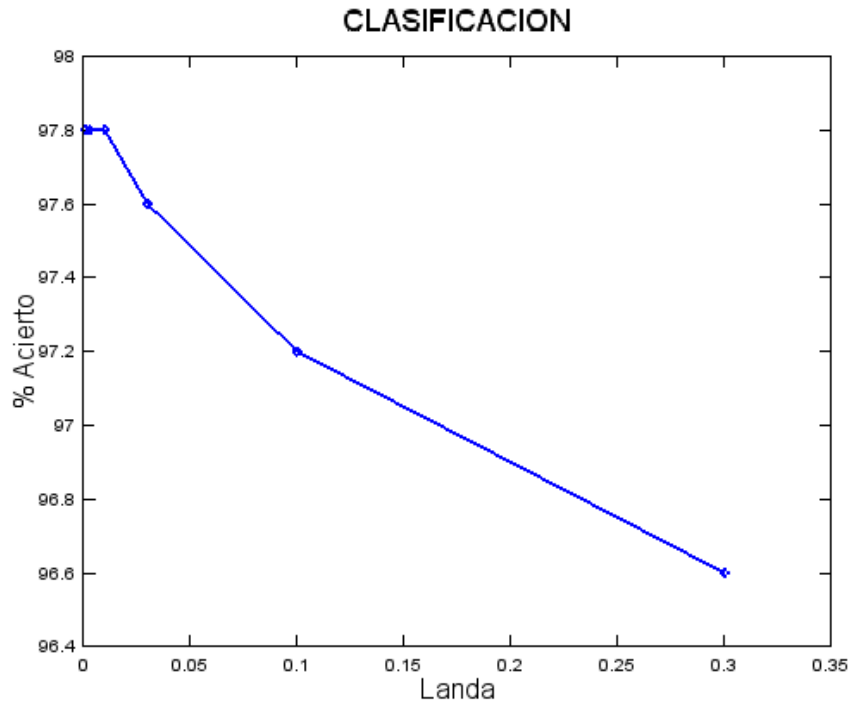
Diferenciar tipos de vino

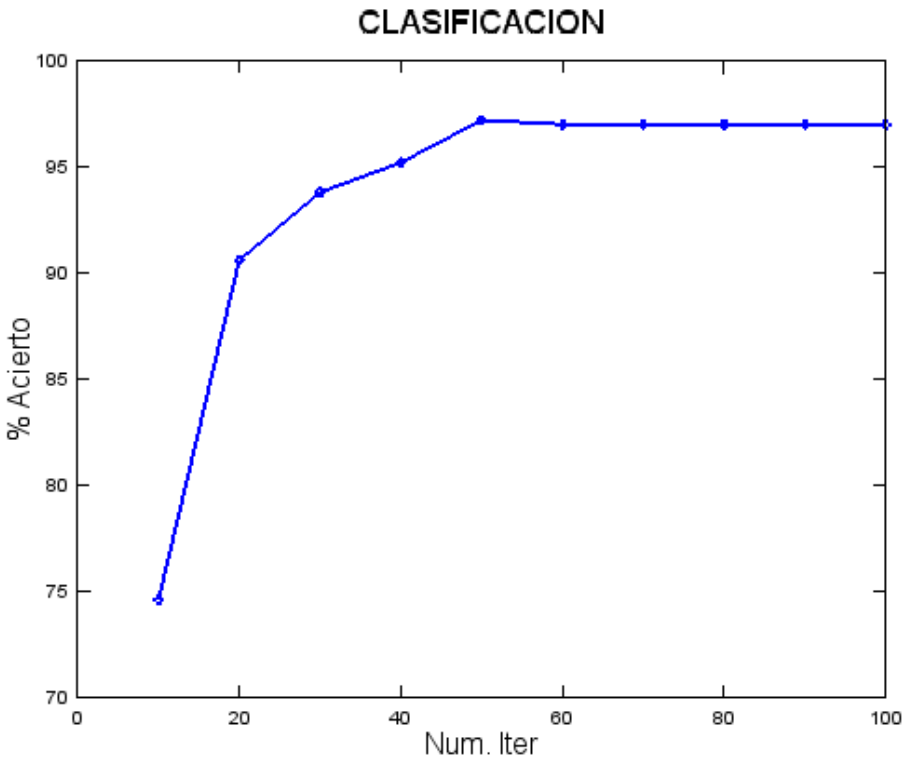
Para ello juntamos todos los ejemplos de entrenamiento, y a los que corresponden a vio blanco le asignamos una y=1 y los de vino tinto y=0.

Después ejecutaremos la regresión logística intentando adivinar a qué tipo de vino corresponde cada muestra.

Lo que se demuestra con estas muestras es que usando las mismas muestras, se consigue una precisión del 97.8%-98%, por lo que es bastante fiable.

Las gráficas obtenidas son las siguientes:





Como podemos observar, el comportamiento del valor Landa es el esperado, obteniendo un máximo en torno a 0.01, y descendiendo rápidamente.

Si tenemos en cuenta el número de iteraciones, vemos claramente que cuanto mayor es el número de iteraciones, mayor es la precisión, estabilizándose en torno al 97% cuando alcanzamos las 50 iteraciones.

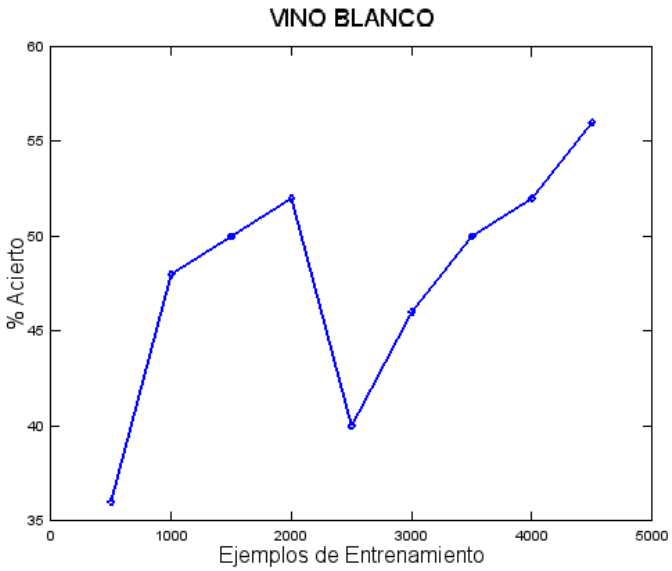
Como conclusión, podemos observar que la precisión para este tipo de problema es similar a la de la regresión lineal, así que podríamos utilizar uno u otro indistintamente.

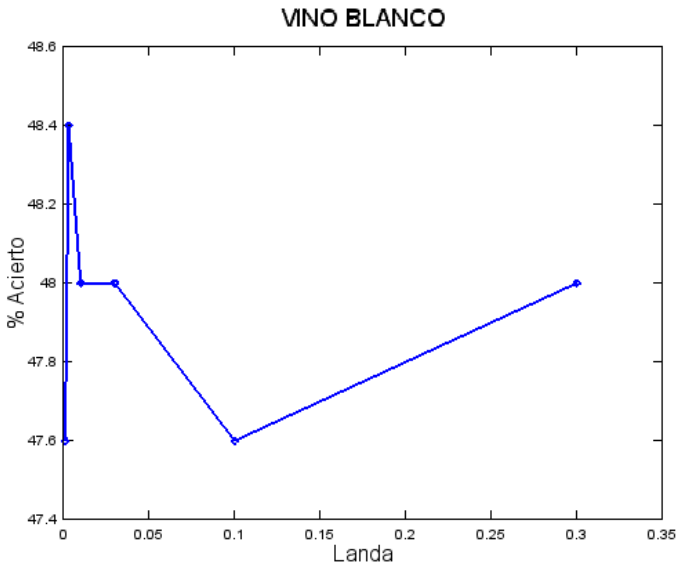
Regresión logística multiclase

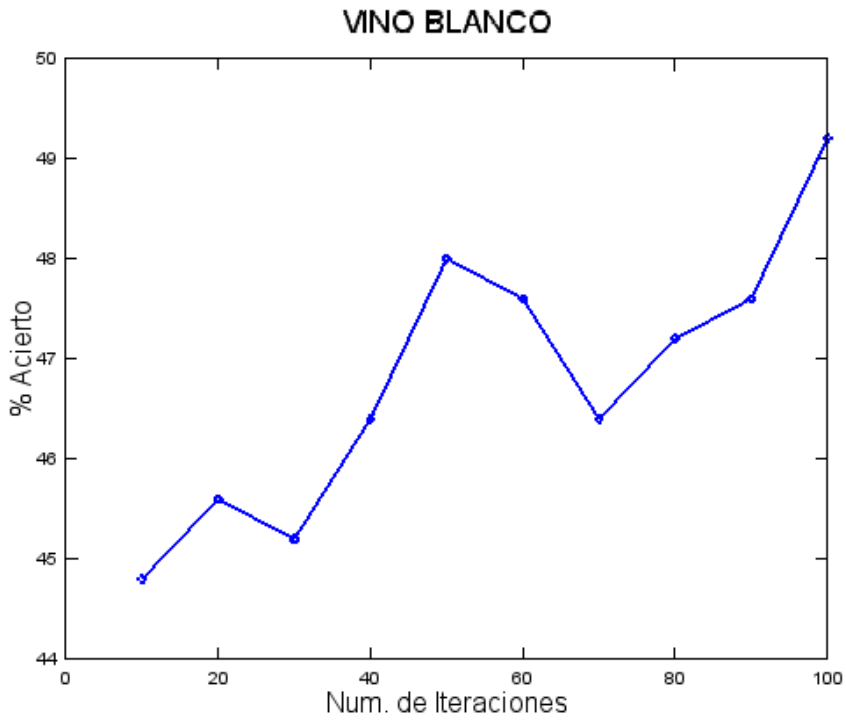
En esta parte vamos a intentar aplicar la regresión logística muticlase para los problemas de adivinar la nota de los vinos:

Vino Blanco

Los resultados de precisión para el problema de adivinar la nota de vino blanco fueron los siguientes:





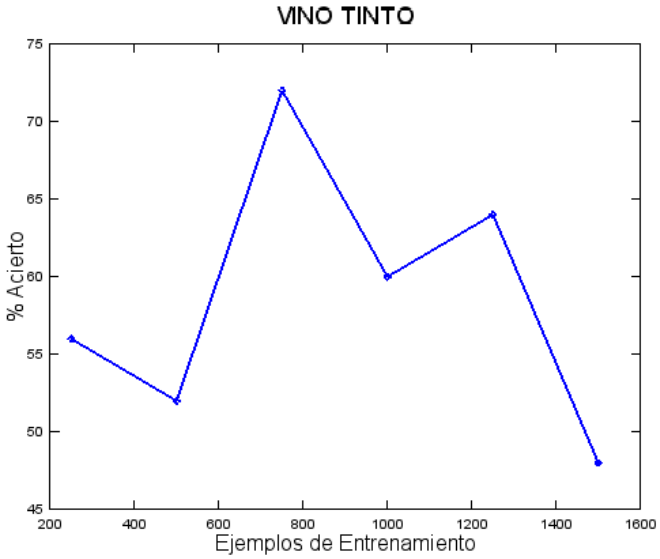


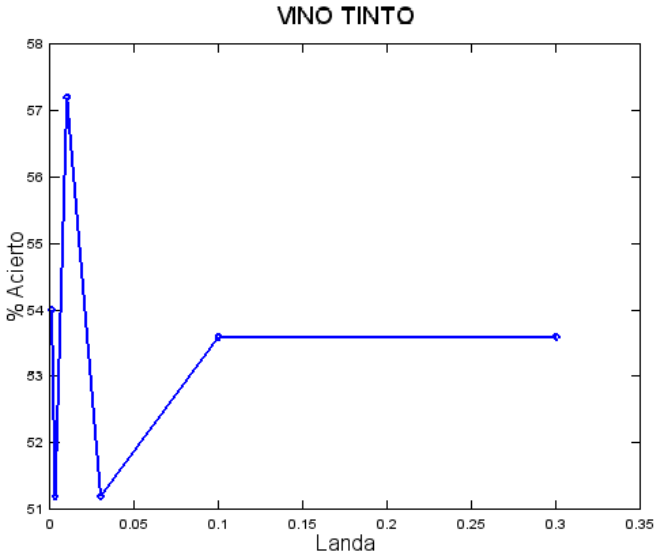
Podemos comprobar que se aprecia un cierto crecimiento en función del número de ejemplos de entrenamiento (aunque hay una gran caída en los 2500, posiblemente porque la elección de los mismos no haya sido muy favorable). Se ve que en el valor de landa, existe un máximo en torno al 0.01%-0.03%, y además podemos apreciar que según aumenta el número de iteraciones aumenta la precisión.

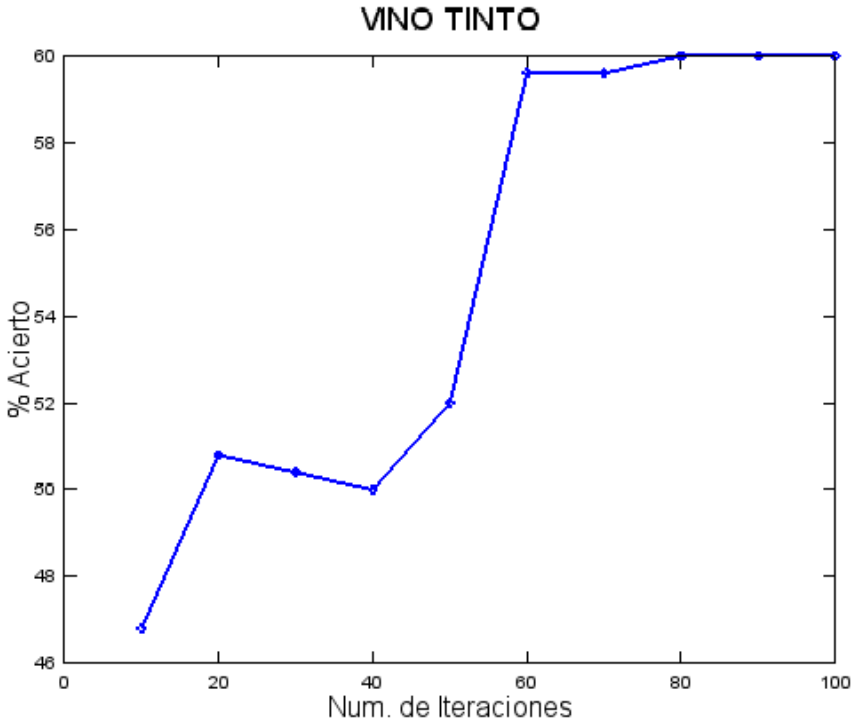
Teniendo en cuenta todo esto, podemos ver que la precisión media se mantiene en torno a 50%-55%.

Vino Tinto

Las gráficas obtenidas para el vino tinto han sido las siguientes:







Podemos observar que el factor landa sigue el mismo funcionamiento, y que mientras el número de iteraciones sigue siendo un factor importante a la hora de aumentar la precisión, el número de ejemplos de entrenamiento no es realmente significativo.

Además, los valores de precisión que se consiguen están en torno al 60%.

Como conclusión, podemos afirmar que este método es algo peor que la regresión lineal para este problema en concreto, aunque no es una diferencia muy significativa (sólo de un 3%-5%).

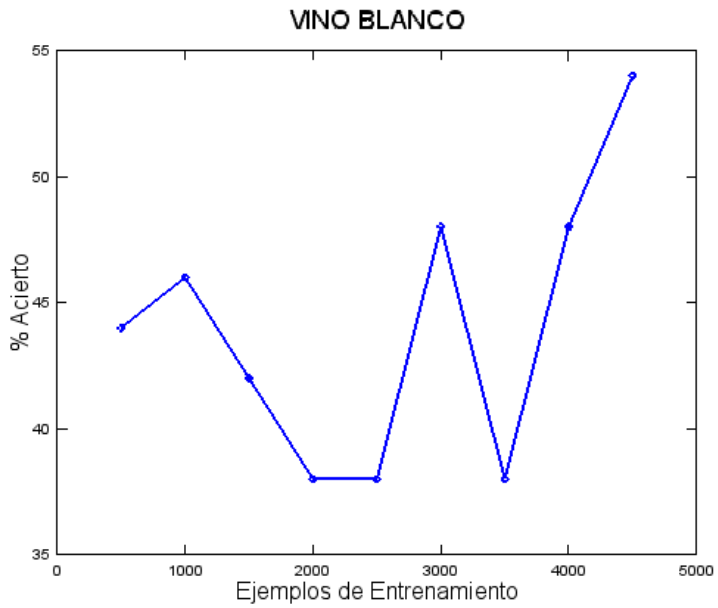
Redes neuronales

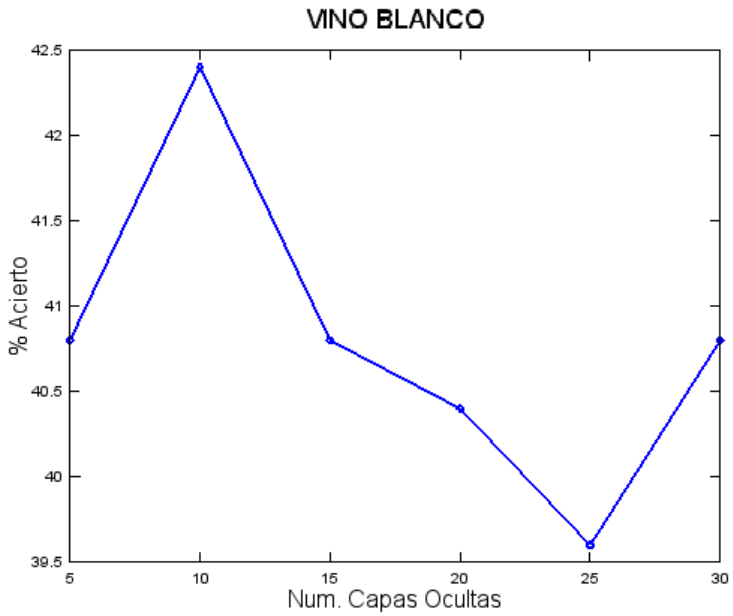
Para aplicar las redes neuronales lo primero que hay que hacer es entrenar la misma para el conjunto de datos de entrenamiento.

Para ello empezamos inicializando los pesos de forma aleatoria.

Vino Blanco

Para el problema de la nota de vino blanco, se han obtenido los siguientes resultados:





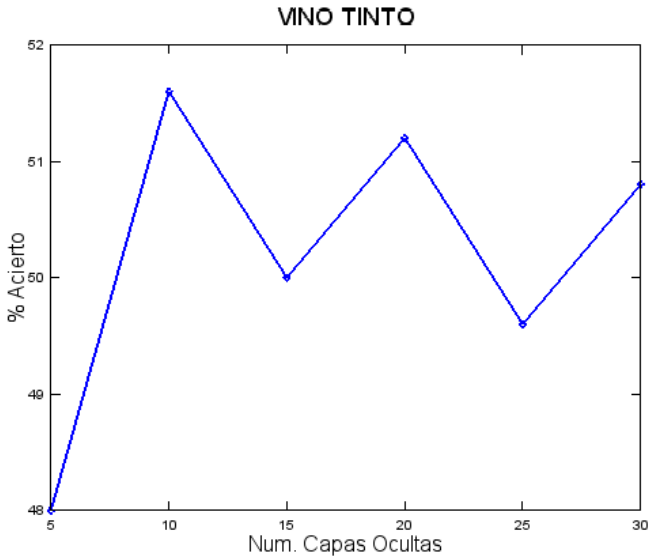
En estos ejemplos podemos ver que a pesar de que aumentan los ejemplos de entrenamiento, no se ve una clara relación de este parámetro con el porcentaje de acierto. Es posible que si el número de ejemplos aumentara considerablemente, si se empezara a ver la relación, pero con tan “pocos” ejemplos de entrenamiento, no se ve la diferencia.

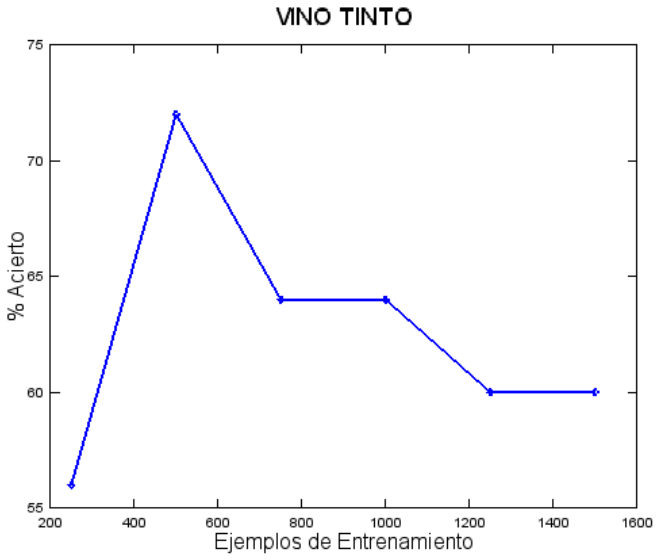
Además, en cuanto al número de capas ocultas, vemos que tenemos un máximo para 10 capas ocultas. Esto puede que tenga relación con el número de atributos del DataSet, que es de 11.

Independientemente de esto, vemos que se observa una precisión de entre un 40%-50%.

Vino Tinto

Los resultados para el vino tinto fueron los siguientes:



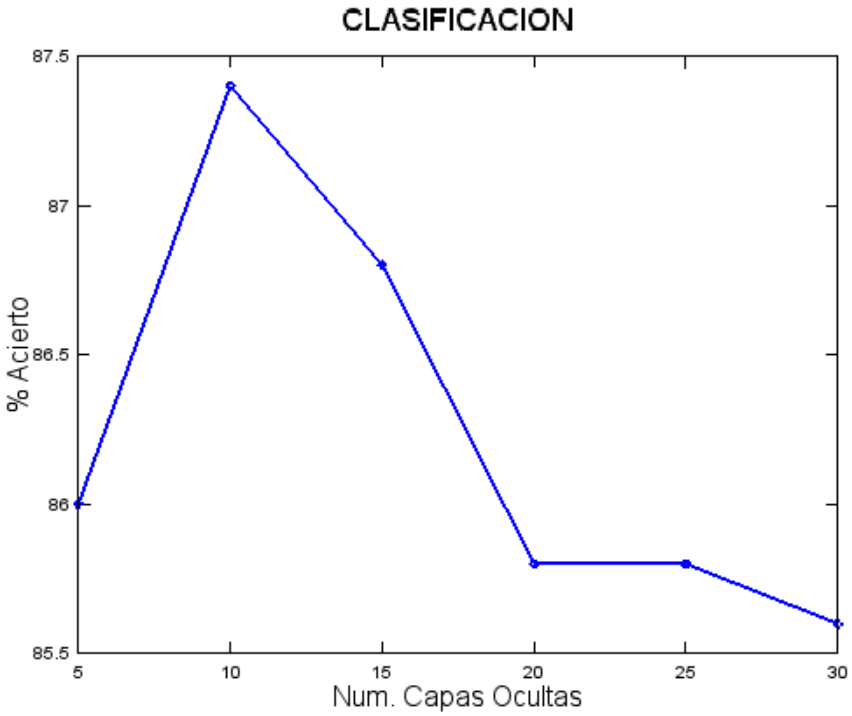


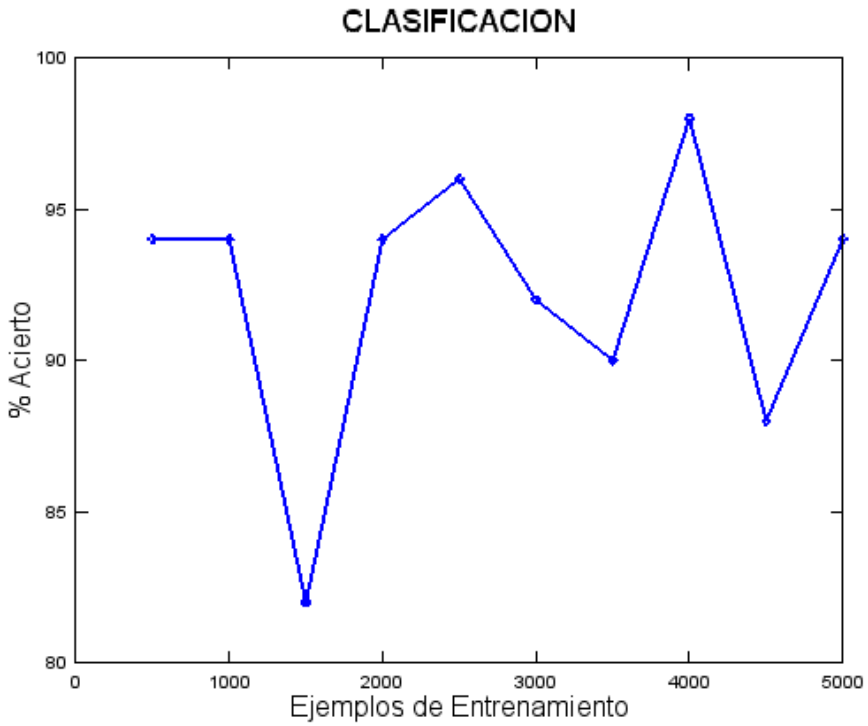
En este caso, vemos que en el número de capas existe un máximo justo en 10 capas (coincide con el caso anterior), y extrañamente, la precisión va disminuyendo según aumentan los ejemplos de entrenamiento.

También se puede ver que la precisión se mantiene en torno a un 50%-60%.

Diferenciar tipos de vino

Los datos obtenidos mediante redes neuronales para diferenciar tipos de vino son los siguientes.





Otra vez, respecto al número de capas, vemos que existe un máximo en 10. Si tenemos en cuenta el número de ejemplos de entrenamiento, no se pueden sacar unas conclusiones muy clarificadoras, pues el resultado va oscilando entre unos valores y otros.

La precisión por este método se mantiene entre 86%-88%, llegando en ocasiones de máximos de 96%.

Como conclusión a las redes neuronales, podemos decir que para el problema de calcular la nota de los vinos las redes neuronales no son una buena solución, pues la precisión que se obtiene es menor que la obtenida con otros métodos.

En cambio, para el problema de diferenciar entre los 2 tipos de vino, los resultados son bastante parecidos a los otros métodos, por lo que usar redes neuronales podría ser una buena opción.

Conclusión

Estos datos se deben a un dataset muy concreto, y posiblemente no muy amplio. Además, las muestras de este dataset no están muy bien equilibradas (esto es que la mayoría de las muestras son de vinos con notas de 5 a 7, mientras que hay muy pocas muestras con notas de 0 a 3 o de 10 a 8). Además no se sabe si todas las variables tienen relevancia en la nota de los vinos, por lo que podría causar desajuste en los resultados.

Quizá sería interesante realizar diversos test para la selección de los atributos más apropiados, aparte de emplear otros algoritmos para detectar los vinos muy malos o muy buenos.

En todos los casos, el aumentar el número de ejemplos de entrenamiento no se correspondía con un aumento de la precisión. Esto se puede deber a la poca simetría de la muestra (falta de valores muy bajos o muy altos).

En cuanto a los resultados de precisión, podemos concluir que para el problema de predecir la nota de un tipo de vino (tinto o blanco), el mejor método de los empleados en la práctica es el de la Regresión Lineal, obteniendo unos resultados del 60%-65% de precisión en el mejor de los casos.

Para el problema de clasificación entre vinos, la precisión de todos los métodos empleados es bastante similar (en torno al 90%-98%), por lo que finalmente descartaremos el método de las redes neuronales, ya que con este conjunto de datos no aumenta la precisión, y el coste en tiempo el muy superior. La Regresión Lineal es casi inmediata, mientras que el entrenamiento de las redes neuronales nos puede llevar varios minutos.

Por último, hemos visto que el aumento en el nº de ejemplos de entrenamiento no mejora la precisión (posiblemente por el desequilibrio de las muestras), y que el nº de iteraciones suele ser muy relevante a la hora de reducir el error.