CUESTIONARIO PRÁCTICA CLASIFICACIÓN AVANZADA: Logistic, SVM, Model Tree, KNN,

Alumno 1 Elena Cantero Molina

Alumno 2 Jezael Pérez Herrera

Grupo 15

1. Vamos a usar de entrada el fichero de datos de enfermedades de soja soybean.arff que consiste en una serie de datos para clasificar enfermedades en la soja
   a) ¿Cuántos ejemplos tiene nuestra base de datos?
      683
   b) ¿Cuántos atributos diferentes tiene nuestra base de datos?
      36

2. Explore los resultados del algoritmo Logistic con la separación training/test de 66 % y con cross validation 10 fold y con n. iteraciones por defecto. Rellene después lo mismo, pero con un número de iteraciones máximo de 1, 2, 10 y 100. El número de iteraciones (MaxIt) se refiere al número máximo de iteraciones en el algoritmo de optimización. El valor -1 es por defecto y no tiene máximo. Haga una tabla con los resultados incluyendo el tiempo en segundos utilizado por el algoritmo para construir el modelo completo que aparece en los resultados. Calcule los márgenes de fiabilidad siempre que necesite comparar sistemas.

	Tasa 66/34 ±	Tiempo 66/34	Tasa 10 fold ±	Tiempo 10 fold
Logistic	*93.5345* ± 1.84 %	*11.87 seg*	*93.8507* ± 1.80 %	*11.49 seg*
Logistic Max It 1	*81.4655* ± 2.91 %	*0.09 seg*	*82.8697* ± 2.83 %	*0.08 seg*
Logistic Max It 2	*88.3621* ± 2.38 %	*0.15 seg*	*84.4802* ± 2.72 %	*0.14 seg*
Logistic Max It 10	*93.1034* ± 1.90 %	*0.65 seg*	*91.8009* ± 2.06 %	*0.64 seg*
Logistic Max It 100	*93.9655* ± 1.79 %	*5.92 seg*	*93.5578* ± 1.84 %	*5.78 seg*

   a) ¿Existe alguna diferencia significativa en los resultados? ¿Cuántas iteraciones mínimas son necesarias?
      Evidentemente, a mayor número de iteraciones, el tiempo de construcción del modelo será mayor. Por otro lado, respecto a la tasa de acierto, podemos apreciar como es necesario un mínimo de 10 iteraciones para obtener unos resultados por encima del 90%.
   b) Comparando los resultados usando 66/34 y 10 fold ¿encuentra diferencias? ¿en qué casos usaría 10-fold?
      En vista de los resultados obtenidos, siempre utilizaría 66/34, a excepción del caso en el que se utiliza una única iteración, donde cross validation tiene una tasa de acierto ligeramente

superior. No obstante, la diferencia más clara entre ambas divisiones es el tiempo que se emplea en la realización de los tests, siendo mucho mayor en el caso de 10 fold, aunque no se vea reflejado en la tabla, algo lógico teniendo en cuenta el funcionamiento del cross validation.

3. Explore los Algoritmos SVM, juegue con Kernels linear, polinomial, RBF, y distintos valores de C. Haga una tabla con los resultados incluyendo el tiempo utilizado por el algoritmo. Rellene solo las celdas en blanco.

	Coste C	Tasa 66/34 ±	Tiempo 66/34	Tasa 10-fold ±	Tiempo 10-fold
Polinomial	1	*13.3621* ± 2.55 %		*15.6662* ± 2.73 *%*	
Polinomial	3	*40.9483* ± 3.69 %		*50.366* ± 3.74 %	
Polinomial	10	*75.8621* ± 3.21 %		*84.3338* ± 2.73 %	
Polinomial	20	*87.931* ± 2.44 %		*91.3616* ± 2.11 %	
RBF	1	*84.9138* ± 2.68 %		*88.7262* ± 2.37 %	
RBF	10	*93.9655* ± 1.79 %	*0.08 seg*	*93.9971* ± 1.78 %	*0.08 seg*
RBF	20	*94.8276* ± 1.66 %	*0.09 seg*	*93.2650* ± 1.88 %	*0.09 seg*
Sigmoid	1	*69.8276* ± 3.44 %		*82.5769* ± 2.84 %	
Sigmoid	20	*93.9655* ± 1.79 %		*93.5578* ± 1.84 %	
Linear	1			*93.9971* ± 1.78 %	
Linear	20			*92.6794* ± 1.95 *%*	
Otros					

a) ¿Son diferentes los resultados máximos en relación al apartado anterior?

No, las diferencias son estadísticamente insignificantes.

b) ¿Existen diferencias entre usar 66/34 y usar 10-fold, especialmente el Sigmoid 1? ¿Qué resultado es más fiable? ¿Puede variar algo en el experimento 66/34 para que la tasa se parezca más a la de 10-fold?

Sí, la tasa obtenida en 10-fold es claramente superior. El resultado más fiable es el del cross validation 10-fold, ya que en este método se realizan hasta 10 pruebas sobre los ejemplos, logrando así un entrenamiento y un testeo de todos los ejemplos del sistema.

Respecto a la última pregunta, si aumentamos ligeramente el valor del coste de los errores, podríamos obtener una tasa de acierto similar a la recogida mediante 10-fold.

c) ¿Influye el coste de los errores*?*

Claramente sí, a mayor coste, mayor tasa de acierto, tanto utilizando la partición 66/34 como con 10 folds.

4. Explore el algoritmo Logistic Model Tree (LMT). Pruebe a variar el mínimo número de ejemplos en cada rama. Analice los resultados y los tiempos

Min. N. ejemplos	Use AIC	Tasa 66/34 +-	Tiempo 66/34	Tasa 10-fold +-	Tiempo 10 fold
15	No	*94.8276* ± 1.66 %	*3.41 seg*	*93.4114* ± 1.86 %	*2.96 seg*
30	No	*94.8276* ± 1.66 %	*2.89 seg*	*93.4114* ± 1.86 %	*3.04 seg*
100	No			*93.4114* ± 1.86 %	*2.78 seg*
1	No			*93.4114* ± 1.86 %	*3.19 seg*

a) ¿Ha cambiado el tiempo necesario y el tiempo de 10-fold?

No significativamente.

b) ¿Influyen el número mínimo de elementos en cada rama en el resultado y en el tiempo? ¿Qué pasaría si tuviéramos muchos más datos?

No. Por tanto, podemos evitarnos la utilización de mayor número de ejemplos ya que no proporcionarán variación alguna.

Si tuviéramos muchos más datos, el resultado sería el mismo teniendo un gasto computacional mayor de forma totalmente innecesaria.

c) ¿Merece la pena ese gasto computacional?

No, ya que los resultados seguirán siendo estadísticamente iguales.

5. Explore el algoritmo 1BK (KNN) con valores de k 1,3,5,10 y ten-fold.

	K				
	1	3	5	10	20
% éxito±margen	91.2152 ± 2.12 %	91.3616 ± 2.11 %	90.1903 ± 2.23 %	87.7013 ± 2.46 %	82.284 ± 2.86 %

a) Explique qué observa en los resultados. ¿existe un óptimo? Razone por qué. ¿Hay diferencias significativas?

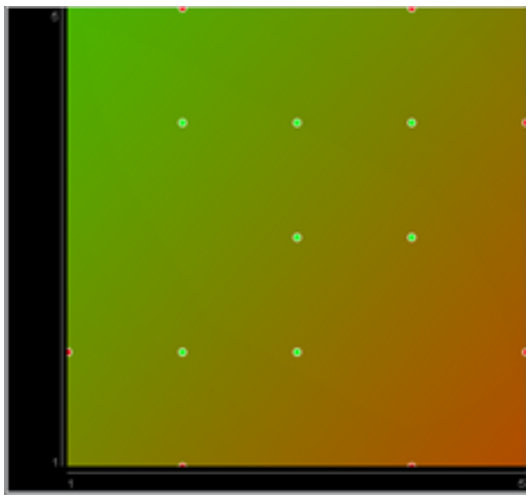
Como podemos apreciar, en el algoritmo 1BK el valor de K óptimo es el 3 y a partir de ahí empieza a disminuir. El valor de K indica el número de ejemplos más cercanos al registro a analizar, de los cuales se extrae la clase mayoritaria. Entonces, existe un valor óptimo para dicho valor, y a partir de él, todos los valores de K superiores irán introduciendo errores.

6. Vamos a comprobar las diferencias entre el algoritmo Logistic y el SVM para el caso de un problema no lineal, tal como dos clases, unos puntos dentro de una elipse y unos puntos fuera de ella. Descargue el fichero ejemplo-svm.arff y ejecute sucesivamente un clasificador logístico y un clasificador SVM

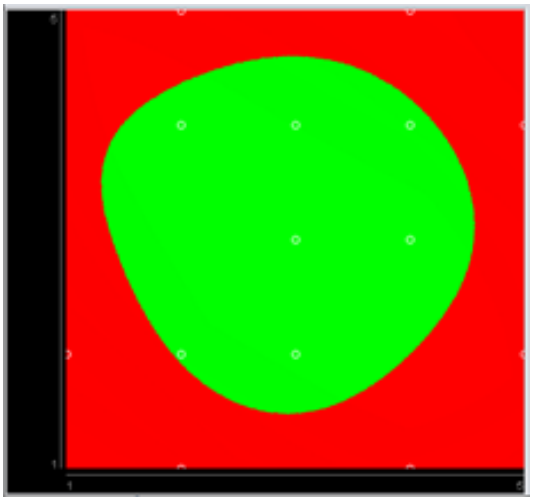
	Tasa 10 folds
Logistic	14.2857 %
LibSVM linear	21.4286 %
LibSVM RBF	64.2857 %

Veamos ahora las fronteras que nos genera el logistic y el LIbSVM RBF. Nos vamos a la pestaña inicial de boundary visualizer. Incluya dos capturas de pantalla

1 Logistic

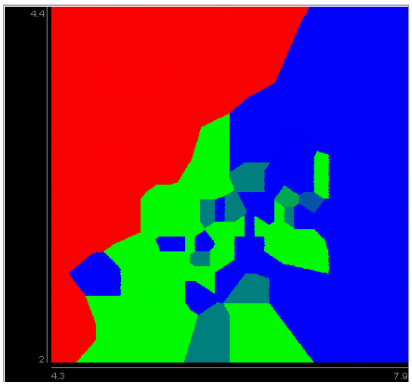


2 LIbSVM RBF

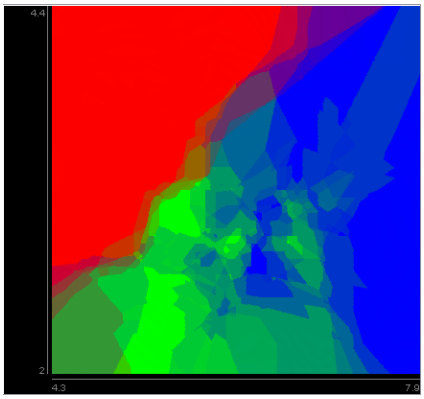


7. Vamos a explorar ahora el tipo de fronteras que nos produce el algoritmo KNN. Para ello abriremos el fichero de datos iris.arff y eliminaremos los atributos petalwidth y petalength usando la opción filter, unsupervised, attribute, remove y guardaremos la nueva base de datos en un nuevo fichero que llamaremos iris2D.arff. Salgamos del explorer y vayamos al boundary visualizer cogiendo el fichero iris2D.arff. Elijamos ahora 1Bk con K=1 y K=5. Saque una copia de las pantallas. Los colores intermedios se producen porque éstos representan una ventana de puntos y los puntos de entrenamiento pueden representar también varios ejemplos superpuestos.

3 K=1



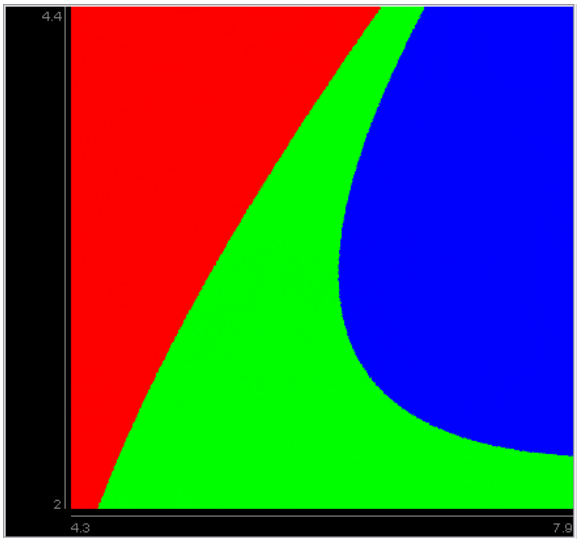
4 K=5



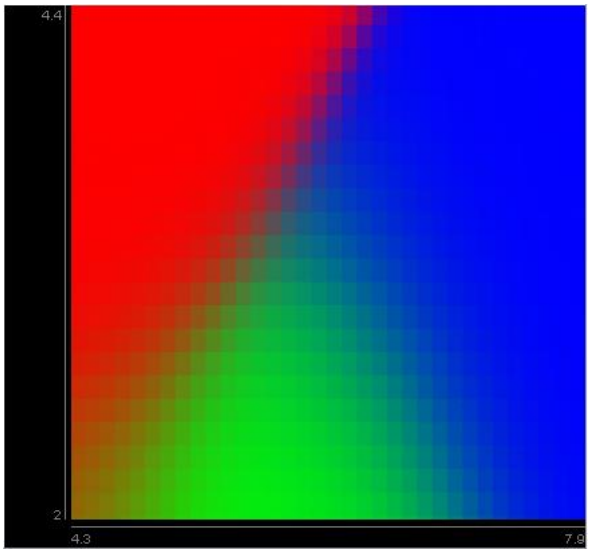
a) Razone las diferencias entre las dos figuras.

Observando la aparición de transparencias en algunas zonas de la gráfica, se puede deducir claramente que el número de errores incrementa significativamente al aumentar el valor de K a 5, con respecto al primer caso con K=1. Esto se debe a que el valor óptimo de K se encuentra más cercano al valor de 1 que al valor de 5.

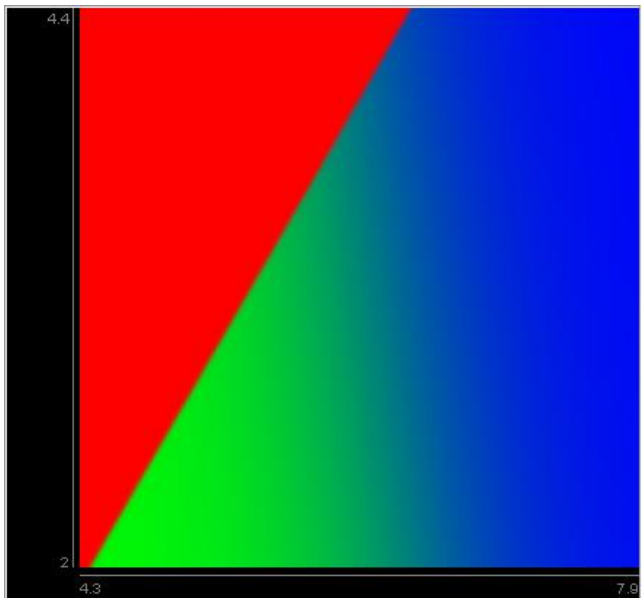
8. Obtenga las fronteras ahora con el algoritmo LibSVM con polynomial



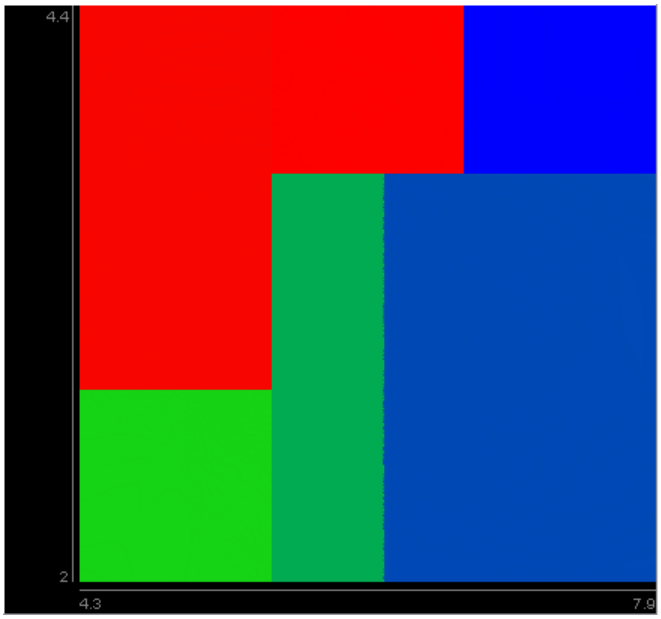
9. Obtenga ahora las fronteras con el algoritmo Naive Bayes



10. Compare ahora con Logistic Regression (Logistic)



11. Compare ahora con el árbol J48



12. Comente las diferencias observadas. ¿Cuáles son lineales y por qué?

Podemos considerar lineales dos funciones. Por un lado, en la función Logistic Regression, la frontera entre el rojo y el verde y azul está representada con una línea recta. Además, la frontera entre el azul y el verde también resulta en una línea recta, aunque se encuentre claramente difuminada debido a la presencia de errores.

Por otro lado, al igual que ocurre en el caso anterior, en la función de Naive Bayes, las fronteras se encuentran representadas por líneas rectas, aunque con poca robustez.

En cambio, en las funciones ilustradas en el apartado 8 y 11, se ve claramente que no son lineales debido a que la tasa de cambio que describen las distintas fronteras presentes en ellas no es constante.