Tarea Final: Análisis de Semillas

Elias Barba Moral
Introducción a la Minería de Datos

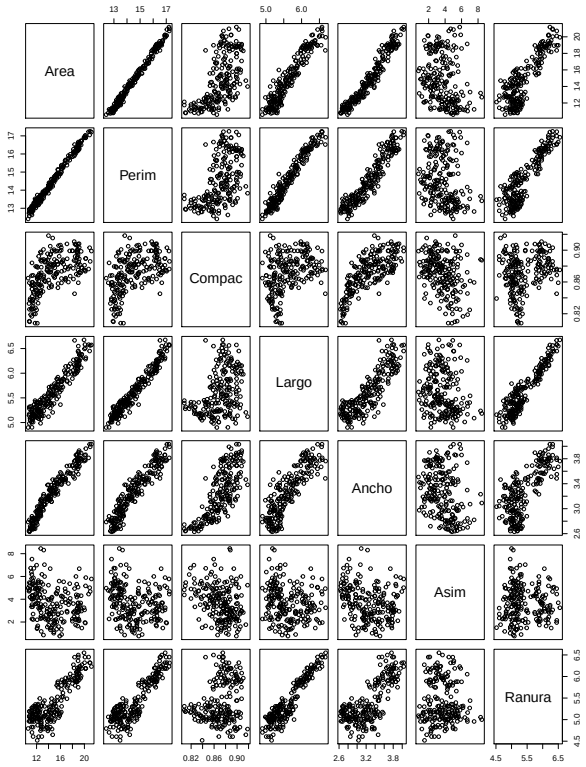
June 6, 2017

1 Introducción

En este trabajo vamos a intentar definir modelos de clasificación para unos datos referentes a semillas de trigo (*https : //archive.ics.uci.edu/ml/datasets/seeds*) . Concretamente tenemos 210 semillas de 3 variedades diferentes con datos completos (sin NA's) referentes a 7 variables relacionadas con el núcleo de cada semilla. El reto consiste en desarollar un modelo que sea eficiente clasificando la variedad de una semilla obtenidas sus variables.
Las variables medidas mediante una técnica de soft X-ray son el área (Area), el perímetro (Perim), lo compacto del nucleo (Compac), la longitud (Largo), el ancho (Ancho), el coeficiente de asimetría (Asim) y la longitud de la ranura del núcleo (Ranura). Todos los parámetros son reales y continuos.
Para empezar el análisis podemos sacar la matriz de correlación de las variables:

	Area	Perim	Compac	Largo	Ancho	Asim	Ranura	Tipo
Area	1.0000000	0.9943409	0.6082884	0.9499854	0.9707706	-0.22957233	0.86369275	-0.34605787
Perim	0.9943409	1.0000000	0.5292436	0.9724223	0.9448294	-0.21734037	0.89078390	-0.32789970
Compac	0.6082884	0.5292436	1.0000000	0.3679151	0.7616345	-0.33147087	0.22682482	-0.53100702
Largo	0.9499854	0.9724223	0.3679151	1.0000000	0.8604149	-0.17156243	0.93280609	-0.25726870
Ancho	0.9707706	0.9448294	0.7616345	0.8604149	1.0000000	-0.25803655	0.74913147	-0.42346287
Asim	-0.2295723	-0.2173404	-0.3314709	-0.1715624	-0.2580365	1.00000000	-0.01107902	0.57727271
Ranura	0.8636927	0.8907839	0.2268248	0.9328061	0.7491315	-0.01107902	1.00000000	0.02430104
Tipo	-0.3460579	-0.3278997	-0.5310070	-0.2572687	-0.4234629	0.57727271	0.02430104	1.00000000

Se puede ver una fuerte correlación entre las variables que tienen que ver con el tamaño (Area, Perim, Ancho, Largo) e incluso con Ranura. Una forma de visualizar mas claramente la matriz de correlaciones es una matriz de gráficos:



Podemos ver que no existen puntos claramente disparatados, o fuera de la tendencia

general. Podemos hacer el mismo dibujo, pero ahora coloreando cada dato con un color referente al tipo de semilla que son:

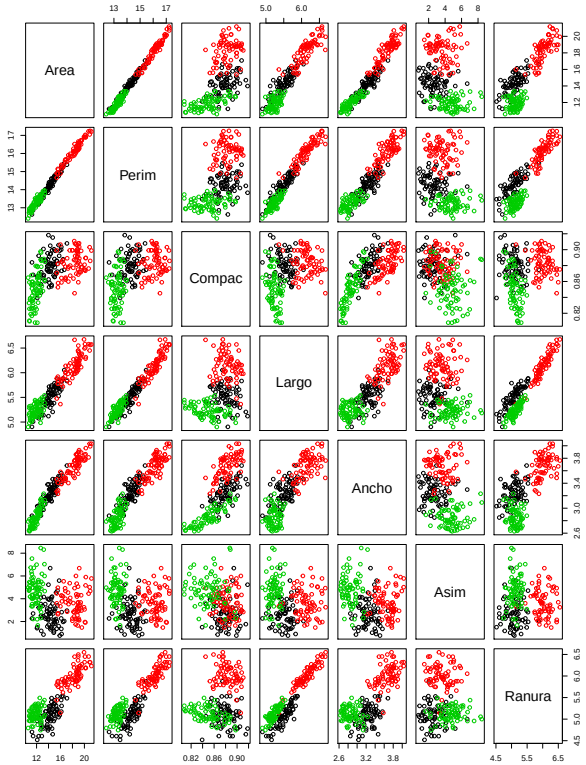
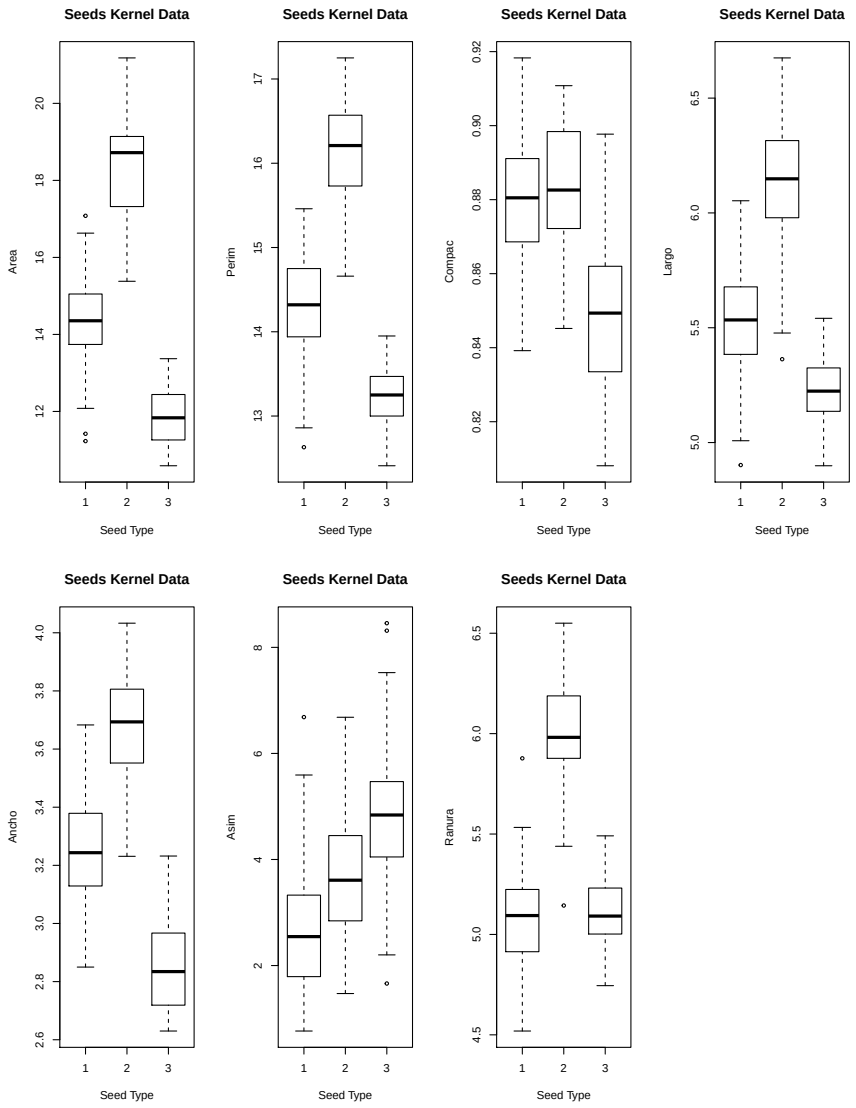


Figure 1: El color negro se corresponde con el tipo 1, el color rojo con el tipo 2 y el color verde con el tipo 3

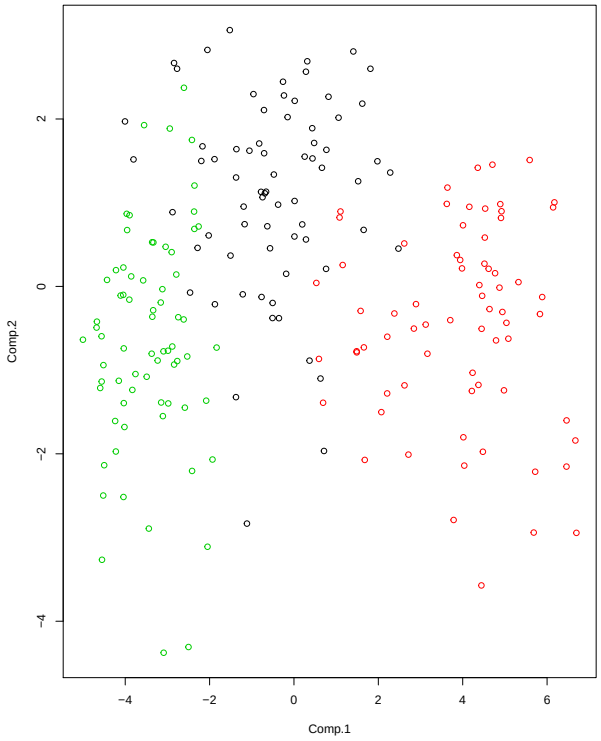
De esta figura podemos concluir que si bien los tipos de semilla no están claramente separados, tienden a estar agrupados, lo cual es un indicador positivo para el éxito de nuestra tarea. En el apéndice A se encuentran los gráficos de densidad de todas las variables, separando por colores el tipo. Otra forma de visualizar los datos es usando boxplots:



De este análisis preliminar podemos ver que variables como Area o Perim son las que mejor diferencian entre las clases, pero no hay ninguna variable que sea determinante o clasifique claramente los tres grupos.
Como hemos visto que hay algunas variables muy relacionadas entre ellas, vamos a realizar un análisis de componentes principales, aunque con la cantidad de datos disponibles, y dado que tampoco hay un número muy grande de variables, no es estrictamente necesario. Además aplicar componentes principales quita interpretabilidad a los resultados.

```
Importance of components:
                          Comp.1    Comp.2      Comp.3       Comp.4       Comp.5       Comp.6       Comp.7
Standard deviation     3.2774884 1.4557867 0.270701706 0.1132524865 0.0522985648 0.0395344339 5.432700e-03
Proportion of Variance 0.8293852 0.1636325 0.005657909 0.0009903061 0.0002111803 0.0001206771 2.278796e-06
Cumulative Proportion  0.8293852 0.9930176 0.998675558 0.9996658637 0.9998770441 0.9999977212 1.000000e+00
```

Esto sirve de confirmación a la idea que las variables están fuertemente correlacionadas, ya que con 2 componentes somos capaces de explicar el 99% de la varianza. Si dibujamos los datos obtenidos en función de estos componentes obtenemos el siguiente gráfico:



Aquí se puede ver claramente que la primer componente, sin quererlo, separa las clases con bastante éxito. Esta componente esta compuesta por la siguiente combinación de variables:

```
Loadings:
       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
Area    0.884 -0.101  0.265 -0.199  0.137  0.281
Perim   0.395        -0.283  0.579 -0.575 -0.302
Compac                                        0.994
Largo   0.129        -0.400  0.436  0.787 -0.113
Ancho   0.111         0.319 -0.234  0.145 -0.896
Asim   -0.128 -0.989
Ranura  0.129        -0.762 -0.613        -0.110
```

Finalmente, para terminar con esta introducción, dividiremos los datos (dado que son muchos) en dos conjuntos: uno para el entrenamiento de los modelos, y otro con el que estimaremos la validez de dichos modelos. Haremos una división del 70% para entrenamiento y un 30% para validación, los datos serán elegidos aleatoriamente, usando la función del paquete "caret": "createDataPartition".

2 Técnicas de clasificación

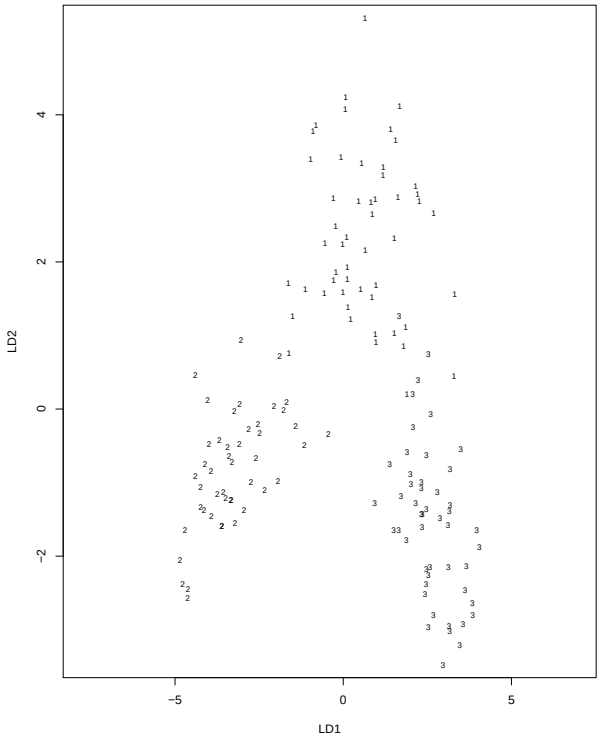
Para llegar a los modelos aquí presentados, han sido comparados con otros modelos similares. Los resultados ofrecidos por dichos modelos pueden ser encontrados en los apéndices B y C.

2.1 Linear Discriminant Analysis

El primer clasificador que vamos a probar es el Linear Discriminant Analysis (LDA). Este clasificador asume que la densidad de las probabilidades condicionales estan distribuidas normalmente, con media y covarianza de cada clase conocida. Adicionalmente se requiere que todas las clases tengan la misma covarianza, osea, se supone homocedasticidad. Con estas condiciones se puede obtener una solución óptima de Bayes, que es nuestro clasificador LDA.
Los resultados de aplicar este clasificador a los datos de entrenamiento son:

```
Call:
lda(Tipo ~ ., data = data_train)

Prior probabilities of groups:
        1         2         3
0.3537415 0.3129252 0.3333333

Group means:
      Area    Perim    Compac    Largo    Ancho     Asim   Ranura
1 14.28962 14.28308 0.8784942 5.508712 3.232173 2.683658 5.094673
2 18.45522 16.19696 0.8825717 6.166435 3.685870 3.707565 6.027957
3 11.76531 13.20816 0.8467469 5.232122 2.831939 4.810306 5.130429

Coefficients of linear discriminants:
               LD1       LD2
Area    0.44417574 -3.583436
Perim  -3.84723400  7.468136
Compac -5.73455116 85.718611
Largo   4.91487096  7.724604
Ancho  -0.09036977 -2.220047
Asim    0.09002501 -0.317333
Ranura -1.74864486 -7.096554

Proportion of trace:
   LD1    LD2
0.6624 0.3376
```

Usando los coeficientes de los discriminantes lineales obtenemos los datos distribuidos de la siguiente manera:



Vemos que los datos quedan bastante bien separados, aunque la separación no es completamente nítida.
Para validar nuestro modelo usaremos el conjunto de datos de validación que tenemos. En primer lugar calcularemos la matriz de confusión:

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 18  0  2
         2  0 24  0
         3  0  0 19

Overall Statistics

               Accuracy : 0.9683
                 95% CI : (0.89, 0.9961)
    No Information Rate : 0.381
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9522
 Mcnemar's Test P-Value : NA
```

Los datos importantes en la figura anterior son la precisión (Accuracy), que nos da una estimación de la probabilidad de clasificación correcta, y la kappa, que usaremos para comparar los modelos. La kappa está definida por la formula $\kappa \equiv \frac{p_0 - p_e}{1 - p_0}$, donde p_0 es la precisión, y p_e es la probabilidad de acertar el tipo de semilla al azar, esto es 33%. Podemos

ver como se distribuyen los datos de validación deacuerdo con la regla creada en la siguiente gráfica:

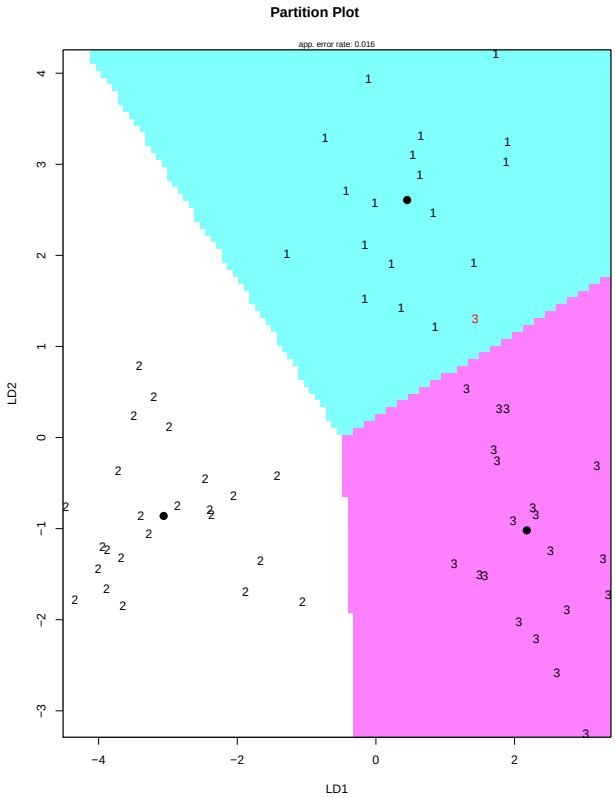
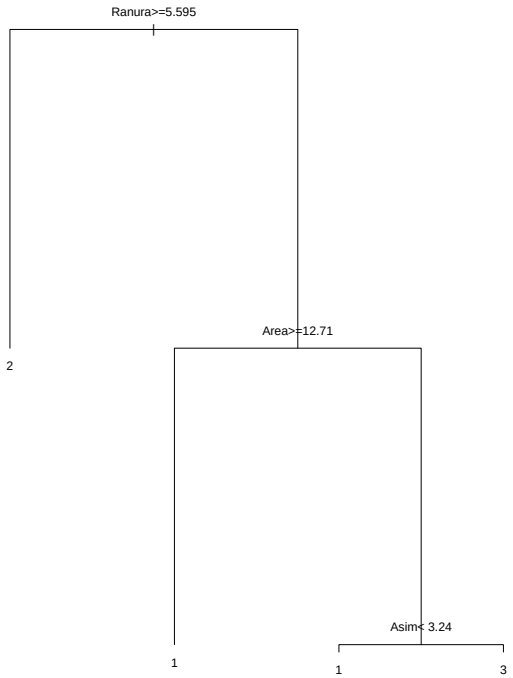


Figure 2: El area blanca corresponde con el tipo 2 de semillas predicha por el modelo. Similarmente, el azul corresponde con el tipo 1 y el morado con el tipo 3. Vemos únicamente un punto de tipo 3 ocupando el area que el modelo predice como tipo 1 como error (en color rojo).

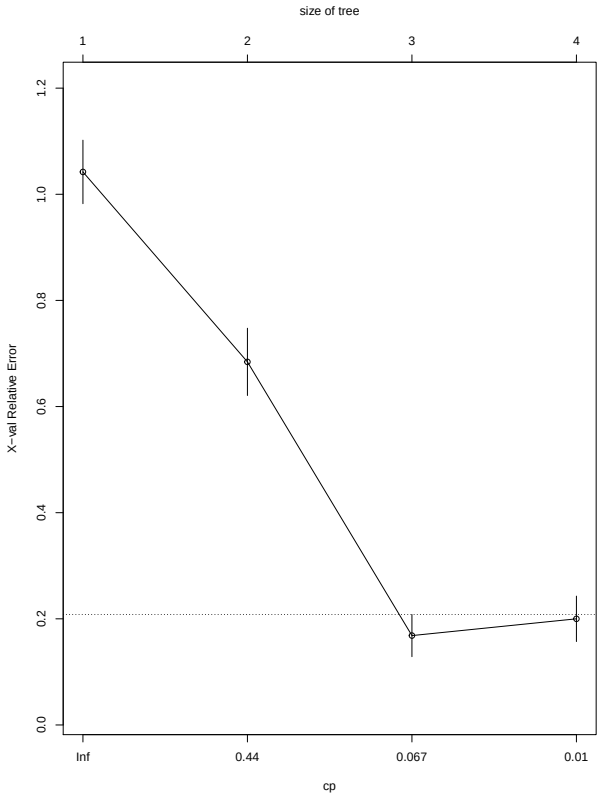
Con esto el modelo queda explicado y evaluado. A continuación pasaremos a presentar un modelo de tipo árbol.

2.2 Árbol de clasificación

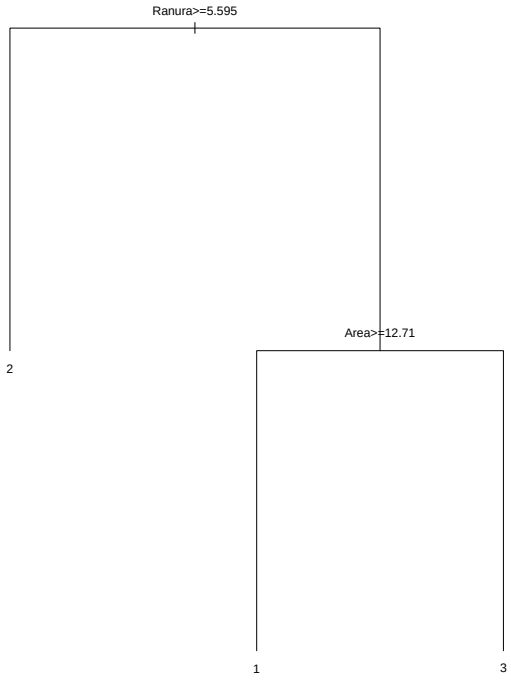
El segundo clasificador que vamos a probar es un clasificador de tipo árbol. En este clasificador se basa en la idea de ir creando divisiones en los datos en determinados valores de las variables hasta llegar a una de las hojas del árbol que se asigna a uno de los tipos de semilla. Introduciendo los datos de entrenamiento en un creador de árboles de R obtenemos:



Este árbol tiene tres nodos (o divisiones) y cuatro hojas. Este modelo ha sido recortado automáticamente, pero aún así deberíamos analizar que tan bueno son dichos recortes. Para ello usaremos el gráfico del error relativo, que consiste en el eje x una evaluación de la complejidad del modelo, y en el eje y la medida del error relativo:



De este gráfico podemos decidir que es mejor tener un árbol un poco mas simple, con tres hojas, con la siguiente forma:



Podemos ver que, usando el conjunto de datos de validación, el árbol comete 5 errores:

```
140 0.02222222 0.97777778 0.00000000
141 0.88461538 0.03846154 0.07692308
142 0.88461538 0.03846154 0.07692308
144 0.10000000 0.00000000 0.90000000
158 0.10000000 0.00000000 0.90000000
163 0.10000000 0.00000000 0.90000000
165 0.10000000 0.00000000 0.90000000
173 0.10000000 0.00000000 0.90000000
174 0.10000000 0.00000000 0.90000000
181 0.10000000 0.00000000 0.90000000
187 0.10000000 0.00000000 0.90000000
188 0.10000000 0.00000000 0.90000000
193 0.10000000 0.00000000 0.90000000
196 0.88461538 0.03846154 0.07692308
197 0.88461538 0.03846154 0.07692308
198 0.88461538 0.03846154 0.07692308
199 0.10000000 0.00000000 0.90000000
```

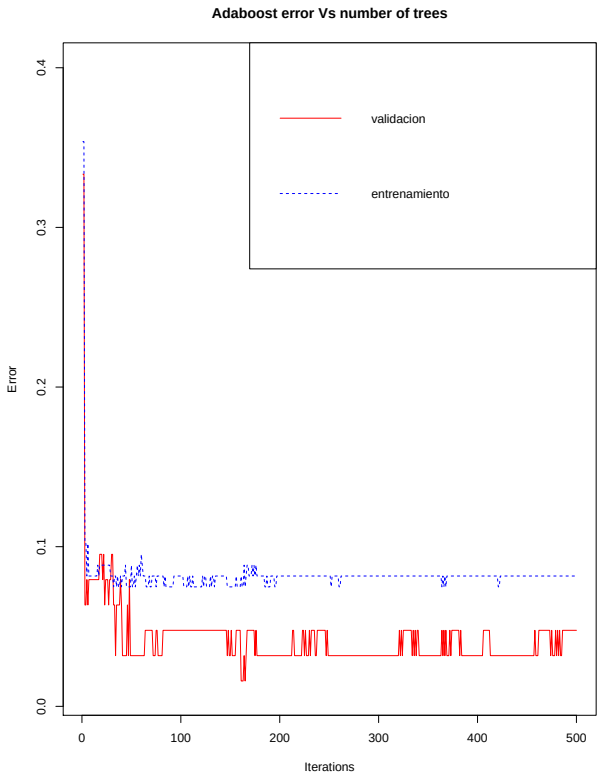
2.3 Boosting

Para mejorar este modelo usaremos la técnica de boosting. Consiste en una mezcla de las técnicas de boostrap (muestreo aleatorio con reemplazo) y la de árboles. Básicamente se crea un árbol, y a partir de él se muestrean los nuevos puntos teniendo en cuenta la probabilidad de estar erróneamente clasificados, de manera que los puntos con mayor probabilidad de

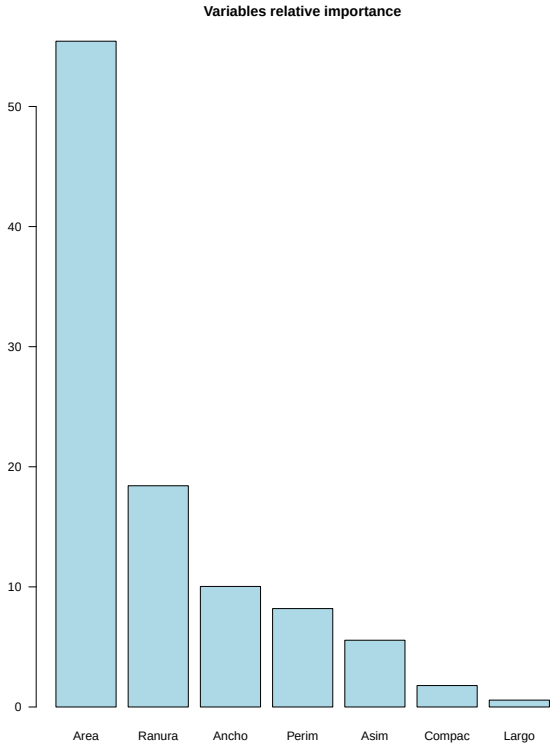
estar mal clasificados son mas relevantes.
Aplicando esta técnica obtenemos la siguiente matriz de confusión:

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 18  0  3
         2  0 24  0
         3  0  0 18

Overall Statistics

               Accuracy : 0.9524
                 95% CI : (0.8671, 0.9901)
    No Information Rate : 0.381
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9283
 Mcnemar's Test P-Value : NA
```

Podemos ver además la evolución del error, tanto del conjunto de entrenamiento como de validación, a mayor número de árboles:



Finalmente podemos ver la importancia relativa de cada variable en la creación de los diferentes árboles creados:

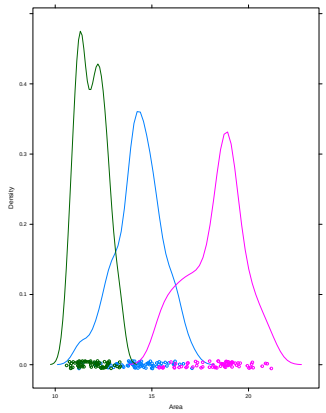

Variables relative importance
50
40
30
20
10
0
Area
Ranura
Ancho
Perim
Asim
Compac
Largo

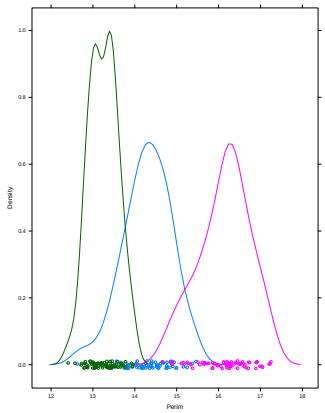
3 Conclusiones

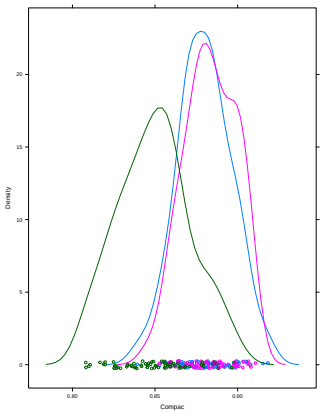
Después de haber analizado los datos y depurado dos modelos, hemos llegados a dos clasificadores que funcionan por encima del 95% en el acierto del tipo de semillas de trigo. Para la validación de estos resultados se separó un conjunto de los datos como datos de validación, que únicamente son usados para evaluar los modelos. En primer lugar hemos conseguido un análisis discriminante lineal, con una precisión del 97% y una $\kappa = 0.9522$. Es un modelo simple pero altamente efectivo. En segundo lugar tenemos un modelo de tipo árbol, que con ayuda de la técnica de boosting es capaz de presentar unos resultados similares, con precisión del 95% y $\kappa = 0.9283$. Este modelo, si bien más complejo, es capaz de demostrar una eficacia similar al anterior.
Ambos métodos funcionan adecuadamente, aunque cabe destacar que, en lo referentes modelos, no siempre modelos más complejos van a dar los mejores resultados.

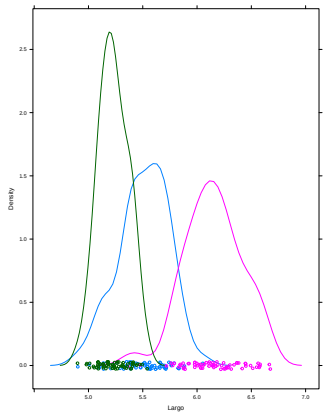
4 Apéndice A: Gráfica de densidad por tipo de semilla y por variable

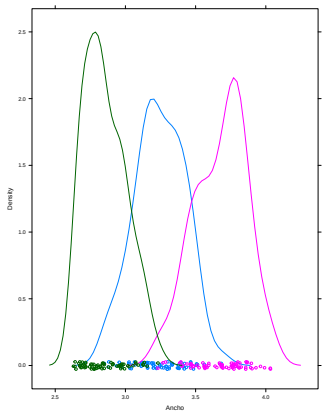
En este apéndice se presentan las gráficas de la densidad de datos por variable separadas por el tipo de semilla. Así el verde corresponde con las semillas de tipo 3, el azul con el tipo 1 y el morado con el tipo 2

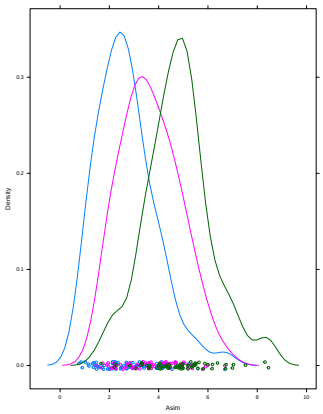


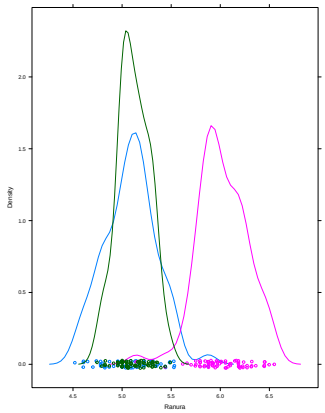












5 Apéndice B: Otros modelos de clasificación

5.1 Quadratic Discriminant Analysis

La fundamentación teórica de este modelo es la misma que para el caso lineal, pero sin asumir homocedasticidad. Los resultados de la matriz de confusión aplicada a los datos de validación:

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 18  0  2
         2  0 24  0
         3  0  0 19

Overall Statistics

               Accuracy : 0.9683
                 95% CI : (0.89, 0.9961)
    No Information Rate : 0.381
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9522
 Mcnemar's Test P-Value : NA
```

5.2 Naive Bayes

Un clasificador del tipo Naive Bayes asume que las diferentes variables son independientes entre si, y la clasificación de un punto viene dada por la combinación de las probabilidades en cada variable del punto de pertenecer a dicha clase. La matriz de confusión al aplicar este método, previamente entrenado con los datos de entrenamiento, sobre los datos de validación es:

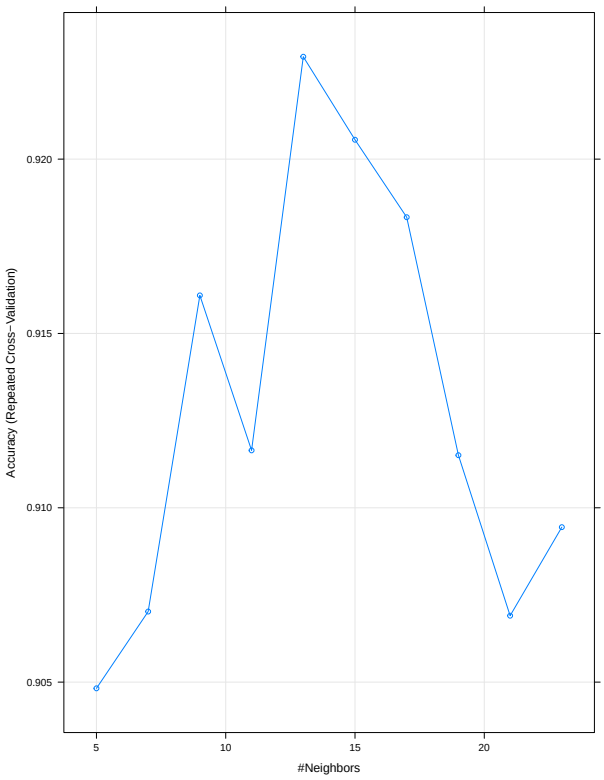
```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 17  1  3
         2  1 23  0
         3  0  0 18

Overall Statistics

               Accuracy : 0.9206
                 95% CI : (0.8244, 0.9737)
    No Information Rate : 0.381
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8805
 Mcnemar's Test P-Value : NA
```

5.3 K-nn vecinos próximos

El método de los K-vecinos próximos consiste en dado un punto que se quiere clasificar, se analizan los k vecinos mas próximos a dicho punto, y se clasifica el punto en la clase que mas vecinos tenga próximos al punto que tratamos de clasificar. En primer lugar se realiza una estimación por validación cruzada de que número de vecinos es mejor para el modelo:

```
k-Nearest Neighbors

147 samples
  7 predictor
  3 classes: '1', '2', '3'

Pre-processing: centered (7), scaled (7)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 133, 133, 132, 133, 133, 132, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.9048214  0.8564286
   7  0.9070238  0.8596044
   9  0.9160913  0.8731813
  11  0.9116468  0.8664777
  13  0.9229365  0.8834618
  15  0.9205556  0.8798811
  17  0.9183333  0.8765478
  19  0.9115079  0.8663143
  21  0.9069048  0.8592685
  23  0.9094444  0.8632653

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was k = 13.
```

Resultando 13 el mejor número:

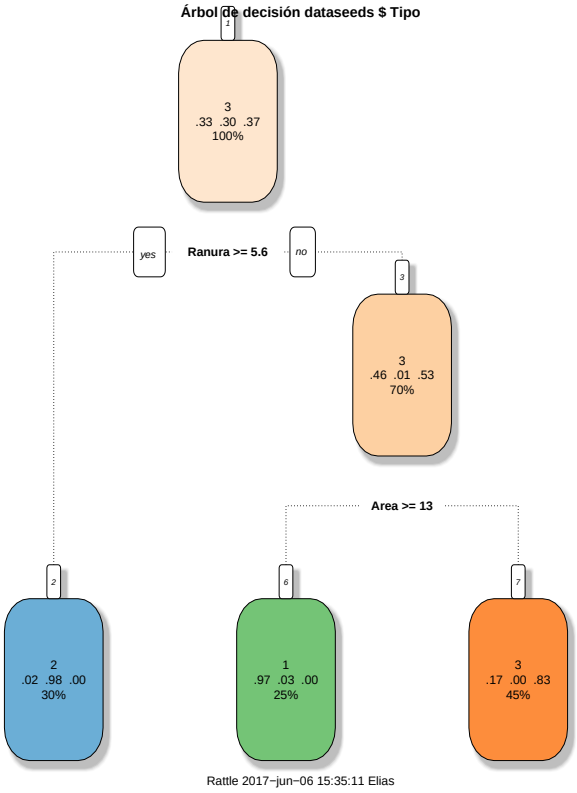


Por tanto, con un modelo de 13 vecinos próximos, con el conjunto de validación obtenemos la siguiente matriz de confusión:

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 17  1  3
         2  1 23  0
         3  0  0 18

Overall Statistics

               Accuracy : 0.9206
                 95% CI : (0.8244, 0.9737)
    No Information Rate : 0.381
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8805
 Mcnemar's Test P-Value : NA
```

6 Apéndice C: Árbol con el paquete rattle

Otra forma de crear árboles de decisión en R es con el paquete rattle. Para hacerlo hay que cargar el paquete, y lanzar la aplicación con el comando "rattle()". Una vez cargado hay que seleccionar los datos que se van a usar en el modelo, que en este caso se selecciona "Conjunto de datos R", y además se selecciona la casilla de partición (70/15/15) con semilla 42. Una vez hecho esto, se le da al botón de ejecutar arriba a la izquierda. Luego se selecciona la pestaña de Modelo y se selecciona el marcador de Árbol. Ahí podemos poner los mismos parámetros que en el árbol que usamos de clasificación anterior y obtenemos un árbol muy similar:


Árbol De decisión dataseeds $ Tipo
1
3
.33 .30 .37
100%
yes
Ranura >= 5.6
no
3
3
.46 .01 .53
70%
Area >= 13
2
2
.02 .98 .00
30%
6
1
.97 .03 .00
25%
7
3
.17 .00 .83
45%
Rattle 2017−jun−06 15:35:11 Elias