Clasificación mediante Máquinas de Vectores Soprte en Frijoles Secos

Alberto Rodríguez

Universidad de Santiago de Chile. https://www.usach.cl/

Abstract. La base de datos "Dry Beans" corresponde a una serie de mediciones obtenidas a distintos frijoles no certificadas como de una única variedad. Se desarrolla un sistema de visión por computadora para distinguir siete variedades diferentes de frijoles secos. Se plantea utilizar el método de clasificación por máquinas de vectores soporte (SVM) a través del paquete "e1071" del software R. Esta investigación concluye con la obtención de que el modelo SVM de Kernel Radial, es la que mejor clasifica a la predicción de la clase e incluso siendo comparado con la técnica de *RandomForest* de la entrega 2. El modelo obtenido tuvo un 98% de precisión, 98% de sensibilidad y 99.7% especificidad.

Keywords: Dry Beans · SVM · Kernel Radial · Kernel Lineal

1 Introducción

El frijol seco (*Phaseolus vulgaris L.*) es la legumbre más importante y producida en el mundo, por lo que desempeña un papel importante en la agricultura de Turquía [1]. La determinación de la mejor semilla es el principal problema para los productores de esta, por lo que es necesario determinar las características físicas como tamaño, color y diversidad. La clasificación manual de estas semillas de frijol es un proceso difícil y que consume mucho tiempo, por lo que se requiere métodos automáticos para clasificar y calificar.

Este artículo tiene la intención de usar el método de clasificación por máquinas de vectores soprte (SVM), el cual permite inferir más información al respecto de los datos contenido dentro del dataset "Dry Beans". Todo esto bajo el objetivo de predecir a cual de las 7 clases pertenece cada frijol, según las variables estudiadas dentro del dataset. Para realizar un contraste y aumentar la precisión del estudio, se elabora un pre-procesamiento de los datos con correlaciones entre los atributos para descartar variables innecesarias y se realiza un equilibrio en cuanto a la cantidad muestral clasificadora de cada clase. Se finaliza esta investigación con una discusión sobre cual de los distintos modelos obtenidos resulta ser el que predice mejor la clase (entre SVM normal, Kernel Lineal, Kernel Linea y Random Forest).

2 Datos de la investigación

Esta conjunto de datos es obtenido por la Facultad de Tecnología de la Universidad de Selcuk. El objetivo del conjunto de datos es predecir según las carac-

terísticas de cada frijol seco si este pertenece a una de las siete tipos de frijoles secos. Se tomaron imágenes de 13.611 granos de las siete variedades, los conjuntos de datos describen características físicas de los frijoles y se tiene una variable resultado, la cual es una variable categórica la cual indica que tipo de frijol se tiene. Estas clases son:

- **Cali:** Es de color blanco, sus semillas son un poco gruesas y un poco más grandes que los frijoles secos y en forma de riñón.
- **Horoz:** Los frijoles secos de este tipo son largos, cilíndricos, de color blanco y generalmente de tamaño mediano.
- **Dermason:** Este tipo de frijoles secos, que son más planos, son de color blanco y un extremo es redondo y los otros extremos son redondos.
- **Seker:** Semillas grandes, de color blanco, la forma física es redonda.
- **Bombay:** Es de color blanco, sus semillas son muy grandes y su estructura física es ovalada y abultada.
- **Barbunya:** Fondo de color beige con rayas rojas o abigarrado, color moteado, sus semillas son grandes, la forma física es ovalada cerca de la redonda.
- **Sira:** Sus semillas son pequeñas, de color blanco, la estructura física es plana.

De los 13.611 frijoles, hay 1.322 (9,7% del total) que pertenecen a la clase Barbunya, 522 (3,8%) a Bombay, 1630 (11,9%) a Cali, 3546 (26%) a Dermason, 1928 (14%) a Horoz, 2027 (14,99%) a Seker y 2636 (19,3%) a Sira.

A continuación se enumeran y se describen los atributos del dataset, los cuales son todos numéricos. Se obtuvieron 12 características dimensionales y 4 de forma (todas medidas en milímetros).

- **Características dimensionales:**
  - **Área:** Zona de frijoles y el número de píxeles dentro de sus límites.
  - **Perímetro:** La circunferencia del frijol o longitud del borde.
  - **Longitud del eje mayor (L):** Distancia entre los extremos de la línea más larga.
  - **Longitud del eje menor (l):** La línea más larga que se puede trazar desde el frijol estando perpendicular al eje principal.
  - **Relación de aspecto:** Defina relación entre L y l.
  - **Excentricidad:** Excentricidad de la elipse que tiene los mismos momentos de la región.
  - **Área convexa:** N° de píxeles en el polígono convexo más pequeño que puede contener el área de la semilla de un frijol.
  - **Diámetro equivalente:** El diámetro de un círculo que tiene la misma área que el área de una semilla de frijol.
  - **Extensión:** La relación entre los píxeles del cuadro delimitador y el área del frijol.
  - **Solidez, Redondez, Compacidad:** Otras medidas del frijol
- **Características de forma:**
  - Shape Factor 1
  - Shape Factor 2
  - Shape Factor 3
  - Shape Factor 4

3 Pre-procesamiento

Tomando en cuenta los valores entregados por el dataset y con el objetivo de mejorar la precisión de la experiencia, se ha realizado un estudio de las variables. Primero se elimina la variable clase para no tener problemas posteriores en el agrupamiento (Este atributo lo utilizaremos después).

Segundo se verifica si el dataset se encuentra completo, es decir, que no hayas datos "perdidos" o algunas variables hayan sido rellenadas con un 0 o un "null". Al usar la función de R *apply*, se pudo verificar que a ninguna variable le faltan datos. Para complementar lo anterior también se busco que no existiera valores extremos (outliers) para esto se gráfico los datos de cada variable (ver **Figura 1**) y ver si existían outliers. Se puedo verificar que no tiene valores extremos.

3.1 Eliminación de variables

Eliminación del Área y Perímetro: A pesar de que la mayoría de los atributos son el resultados fórmulas que contienen como entrada otros atributos, es decir, atributos son dependientes de otros, estas son variables aleatorias y es importante ver si existe algún tipo de relación entre estas. Para esto evaluarlos utilizaremos una matriz de correlación.

Como se aprecia en la **Figura 2** existen muchas variables con varias relaciones entre ellas (Los colores azul y rojo mas oscuro indican que existe mayor relación), el coef. de correlación utilizado es Pearson ya que se trata solo de variables numéricas. Se puede observar que existe una relación muy grande entre el área y el perímetro, además estas tiene una relación muy grande con otras variables, así que se decide eliminar ambas

Eliminación de 4 variables más: Al realizar nuevamente una matriz de correlación (Ver **Figura 3)** Se puede apreciar que las variables "Compactness" y "Shape Factor3" tiene una fuerte relación cal igual que ConvexArea" y "Equiv-Diameter", por lo que se realiza un test de hipótesis para ambos, en donde:

- **H0:** No hay relación entre las variables
- **H1:** Hay relación entre las variables

Para realizar el test, se realiza un estudio del coef. de correlación de Pearson con 95% de confianza, el cual se resume en la Tabla 1:

Table 1. Test de correlación entre Compacidad y Shape Factor 3.

Atributos	**Correlación**	**P-valor**	**Intervalo de confianza**
Compacidad / ShapeFactor3	0.9986	$< 2.2e\text{-}16$	[0.99864 0.99872]
ConvexArea / EquivDiameter	0.9852	$< 2.2e\text{-}16$	[0.98472 0.98571]

Con los resultados se cumple H1 y las cuatro variables esta relacionadas por lo que se eliminan.

Eliminación de Longitud del eje menor y ShapeFactor1: Se realiza nuevamente una matriz de correlación (Ver **Figura 4**) y se encuentra relación entre los atributos "MinorAxisLength" y "ShapeFactor1" y se realiza el mismo test de hipótesis anterior obteniendo resultados similares, donde se cumple H1 y se eliminan ambas variables.

El último paso es balancear las clases, dejamos aproximadamente 1000 datos de cada clase (Excepto bombay que son en total 522).

4 Método Utilizado

El método utilizado para realizar la clasificación de los datos existentes dentro del dataset corresponde a Máquinas de Vectores Soporte (SVM), esta se origina con el objetivo de poder solucionar problemas de optimización no lineales con restricciones de desigualdad, la cual permite separar las clases del problema utilizando funciones Kernel. Estas permiten realizar separaciones lineales y no lineales, logrando modificar el grado de curvatura para estos últimos siendo denominada como *gamma* y el costo equivale al rango entre los vectores soporte.

5 Resultados de la investigación

En esta sección se presenta la obtención de un modelo SVM sin hacer uso de la función tune, seguido por un modelo SVM con kernel lineal y función tune, finalizando con el modelo SVM con kernel radial y función tune. Es importante notar que para generar los modelos antes mencionados, se ha utilizado el conjunto de datos obtenido en la etapa de pre-procesamiento.

Para el caso de los modelos obtenidos por medio de un kernel lineal y radial, se ha utilizado la técnica de Validación Cruzada, esto con el objetivo de comprobar independencia entre los datos de entrenamiento y prueba obtenidos en el dataset. Respecto a la grilla de valores de costo y gamma, se han implementado 2 rangos en cada kernel (ver **Tabla 2**); sin embargo, en esta sección solo se abordaran correspondientes al rango 2, ya que los resultados obtenidos con el rango 1 no varían mucho respecto al 2. El error y performance del modelo son iguales.

Valores	**Rango 1**	**Rango 2**
Costo	$[2^1, 2^3]$	$[2^{-1}, 2^5]$
Gamma	$[2^{-4}, 2^5]$	$[2^{-2}, 2^4]$

Table 2. Tabla de rangos de costo y gamma utilizados en el kernel.

5.1 Modelo sin cambios (sin función tune)

Se obtiene el modelo SVM a partir de un conjunto de entrenamiento de datos (con un 70% del dataset) y de test (30% del dataset). Los vectores soporte obtenidos

son 680, mientras que tiene un 95.8% de precisión, 94% de sensibilidad y 99% especificidad. La matriz de confusión en la **Tabla 3** del anexo, resume los datos mencionados anteriormente.

En la **Figura 5** se muestran los vectores soporte representados con el símbolos "+", además de las siete clasificaciones realizadas por el modelo, diferenciadas por distintos colores.

5.2 Modelo Kernel Lineal con función tune

A partir del rango 2 (sin considerar el valor gamma) presente en la **Tabla 2**, el mejor modelo generado tiene un costo de 2 (ver **Figura 6**) y 723 vectores soporte. Para este caso, la performance obtenida es de 0.0308.

En la matriz de confusión (ver **Tabla 4**), se puede apreciar la clasificación realizada por el mejor modelo obtenido por medio de un kernel lineal. Para este caso, se tiene un 97.2% de precisión, 97% de sensibilidad y 99.2% especificidad.

En la **Figura 7** se puede observar los vectores soporte representados por el símbolo "+", además de las siete clasificaciones.

5.3 Modelo Kernel Radial con función tune

Al igual que con kernel lineal, se hace uso del rango 2 presente en la **Tabla 2** para la obtención del mejor modelo. Dicho modelo es generado a partir de un costo 8 y un valor gamma 0.25 (ver **Figura 8**), donde se hace uso de 1011 vectores soporte y la performance obtenida es 0.032.

En la **Tabla 5** es posible ver el detalle de la mejor clasificación realizada por el modelo obtenido. De acuerdo a sus valores, se tiene un un 98% de precisión, 98% de sensibilidad y 99.7% especificidad.

Al igual que con kernel lineal, la **Figura 9** muestra los vectores soporte representados con el símbolo "+" y las siete clasificaciones.

6 Discusión

El modelo SVM sin utilizar tune, tiene la función de entregar información sobre el contexto de estudio. Sus resultados permiten establecer una base porcentual sobre los valores adecuados para clasificar, donde se puede ver que para un costo 1 y gamma 1 la cantidad de semillas clasificadas adecuadamente es de un 95.8% (presición). Es un valor muy alto y permite intuir que con un kernel lineal o radial, los porcentajes de precisión, sensibilidad y especificidad deberían aumentar.

Para la obtención de la grilla adecuada, se han utilizado 2 rangos. Como se dijo anteriormente, el performance y los resultados no eran muy distintos entre ambos rangos, pero se decidió usar el rango 2 ya que era un mínimo mejor la precisión que el rango 1, tanto para el kernel lineal como el radial, donde se han clasificado un mayor número de semillas correctamente, específicamente, el costo 2 para el kernel lineal y el costo 8 y 0.25 para el kernel radial.

En cuanto al mejor modelo, se observa que los 3 modelos son muy buenos obteniendo todos una precisión sobre el 90%, aunque las diferencias no varían mucho entre ellos el mejor es el implementado con el kernel radial. Este método es demasiado bueno ya que además tiene una sensibilidad y especificidad de 98% y 99.7% respectivamente. Para analizar si cambiando las iteraciones de la validación cruzada esto podía mejorar, se hicieron pruebas con un cross de 2,3 y 10 respectivamente, obteniendo prácticamente el mismo resultados en todos los casos, pero se optó por usar el cross igual a 2, ya que el algoritmo demoraba menos en ejecutar. Además indicar que para esta experiencia no se puede usar la curva ROC, ya que es una representación gráfica de la sensibilidad frente a la especificidad para un clasificador binario.

Respecto a la importancia de las variables y su respectiva eliminación de alguna de estos, no se pudo utilizar la librería *RWeka* por problemas técnicos, por lo que me orienté de la importancia de las variables de la experiencia 2 (ver **Figura 10**). Al eliminar la variable "Extent" en ambos métodos (lineal y radial) los resultados no cambian en nada, pero al eliminar la variables "Solidity", ambos métodos empeoran su precisión a menos del 90% por lo que se decidió eliminar la variable redundante solamente y dejar las demás, tratando de cumplir el principio de parsimonia.

Comparando el modelo kernel radial obtenido por medio de SVM y el modelo obtenido con Random Forest, se puede decir que el primero es mejor respecto a la clasificación del problema. Aunque SVM aumenta solo en un 3% mas la precisión, esto es porque ambos estan sobre el 90% por lo que no se puede aumentar mucho más.

7 Conclusiones

Para esta investigación se realizaron distintos modelos de clasificación utilizando el método SVM, gracias al uso de la librería "e1071" proveniente de R, al igual eliminación de datos NA y una selección de variables, que permitieron mejorar la calidad de los distintos modelos generados, así como también, se variaron los parámetros de kernel, costo y curvatura.

De los modelos obtenidos, se puede mencionar que la clasificación de las distintas semillas, es mejor cuando se realiza con la técnica SVM radial, comparado con la lineal y con el obtenido por Random Forest, aunque esta mejora es pequeña, a pesar del leve aumento de error obtenido, esto se debe a que el método SVM permite acomodar una separación de clases mayormente determinista.

Se concluye esta investigación mencionando que la teoría indica que la disminución en la cantidad de variables facilita la clasificación de un modelo, como ocurrió con el método de Random Forest para este dataset y las baja importancia de algunas variables dadas por "GINI". Algo similar sucedió con el método SVM, donde la eliminación de la variables en el pre-procesamiento permitió mejorar la clasificación no así, como cuando se analizó la importancia de cada una, donde solo se encontró una variable redundante pero no mejoraba la clasificación.

References

1. Koklu, M., Ali Ozkan, I. (2020, mayo). Multiclass classification of dry beans using computer vision and machine learning techniques. https://www.sciencedirect.com/science/article/pii/S0168169919311573?via

8 Anexos

A A Análisis Estadístico

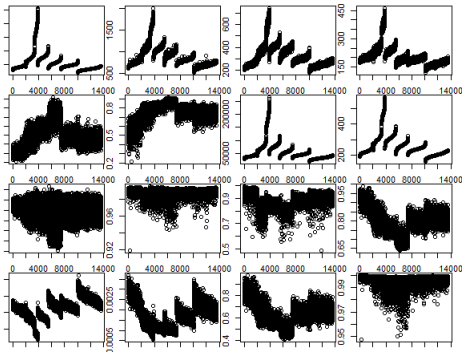

Fig. 1. Busqueda de Outliers en los datos de las variables

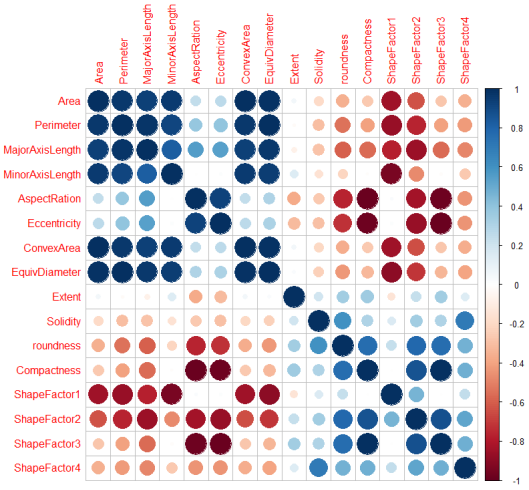

Fig. 2. Matriz de correlación con todas las variables

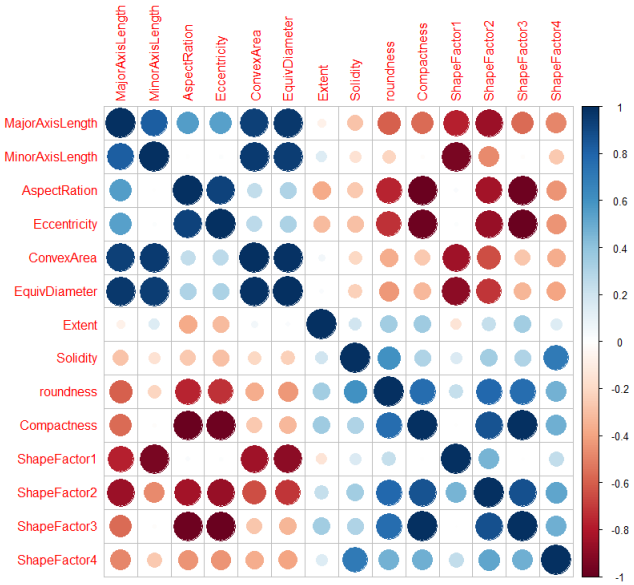

Fig. 3. Matriz de correlación con todas las variables

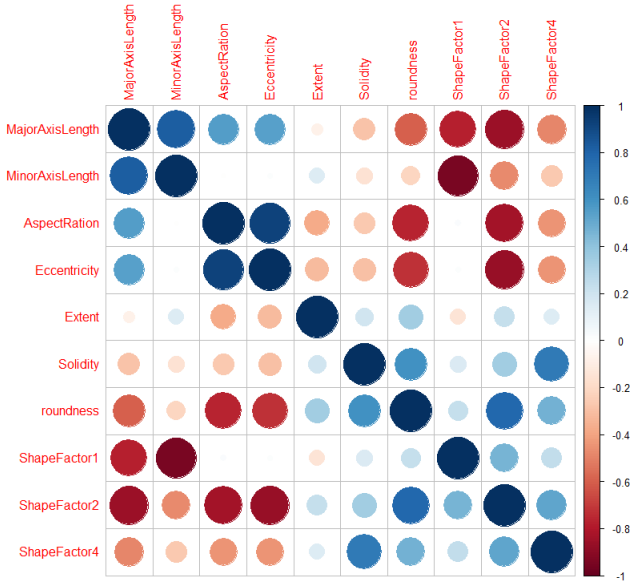

Fig. 4. Matriz de correlación con todas las variables

B B Aplicación Método SVM

Table 3. Matriz SVM sin función tune

Clases	Barbunya	Bombay	Cali	Dermason	Horoz	Seker	Sira
Barbunya	676	0	12	0	3	1	1
Bombay	0	348	0	0	0	0	0
Cali	29	3	706	0	6	0	0
Dermason	0	0	0	710	0	34	0
Horoz	4	0	7	0	696	0	14
Seker	1	0	0	26	0	659	12
Sira	6	0	0	9	2	25	667

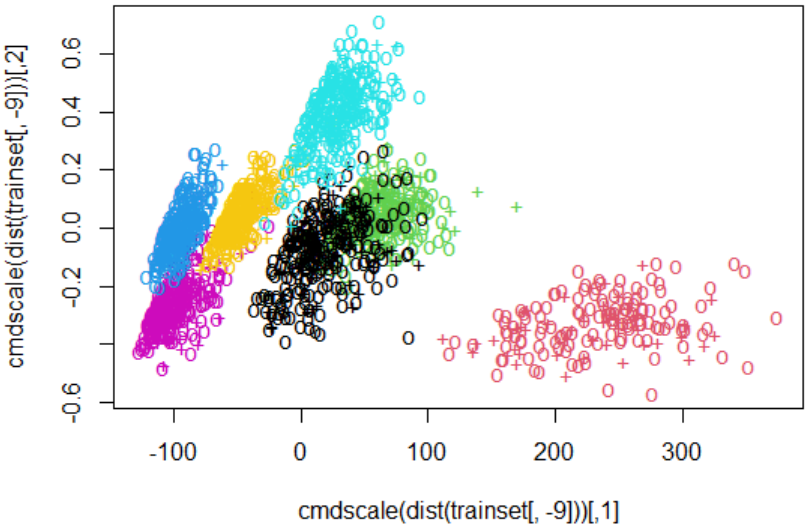

Fig. 5. Gráfico SVM sin función tune

Table 4. Matriz SVM con Kernel Lineal

Clases	Barbunya	Bombay	Cali	Dermason	Horoz	Seker	Sira
Barbunya	990	0	23	0	5	0	0
Bombay	0	522	0	0	0	0	0
Cali	28	0	1004	0	6	0	0
Dermason	0	0	0	1024	0	20	0
Horoz	4	0	3	0	1006	0	13
Seker	0	0	0	22	0	975	17
Sira	0	0	0	0	11	32	1006

Table 5. Matriz SVM con Kernel Radial

Clases	Barbunya	Bombay	Cali	Dermason	Horoz	Seker	Sira
Barbunya	1002	0	12	0	2	0	0
Bombay	0	522	0	0	0	0	0
Cali	17	0	1018	0	2	0	0
Dermason	0	0	0	1022	0	21	0
Horoz	3	0	0	0	10—8	0	10
Seker	0	0	0	24	0	984	11
Sira	0	0	0	0	0	22	10—5

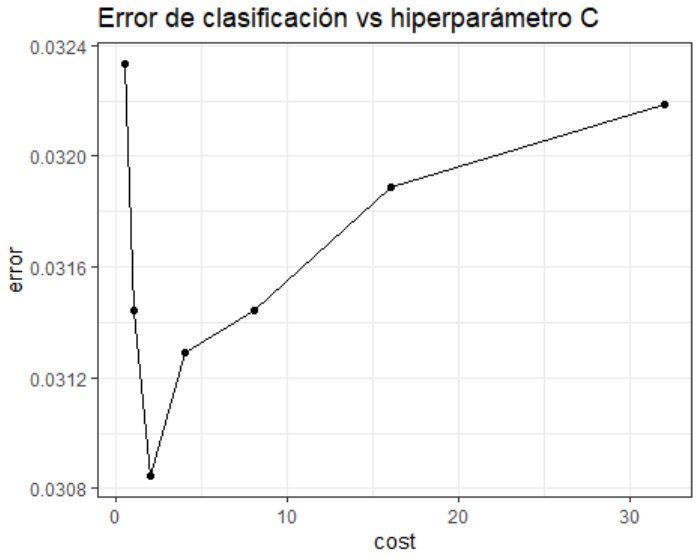

Fig. 6. Error clasificación SVM con Kernel Lineal y rango 2

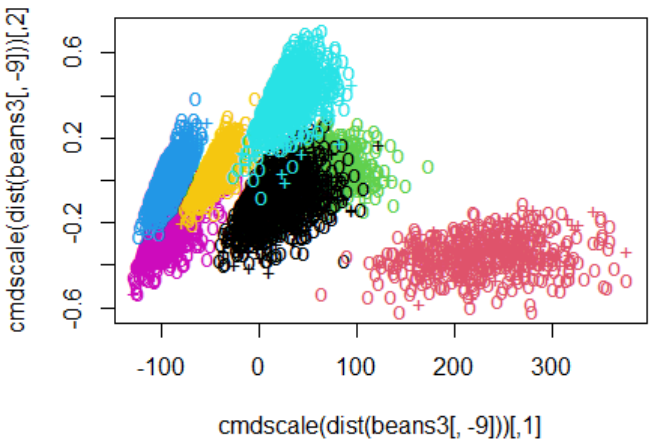

Fig. 7. Gráfico SVM con kernel lineal y rango 2

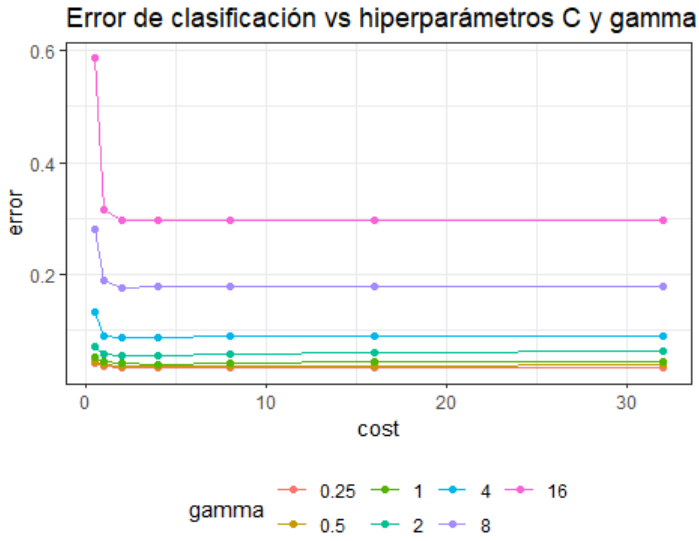

Fig. 8. Error clasificación SVM con Kernel Radial y rango 2

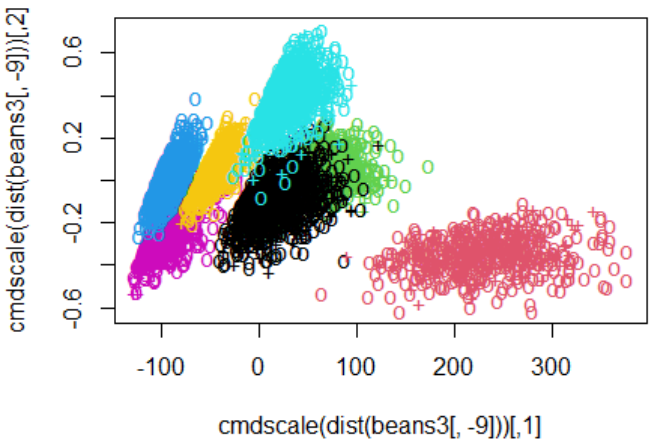

Fig. 9. Gráfico SVM con kernel radial y rango 2

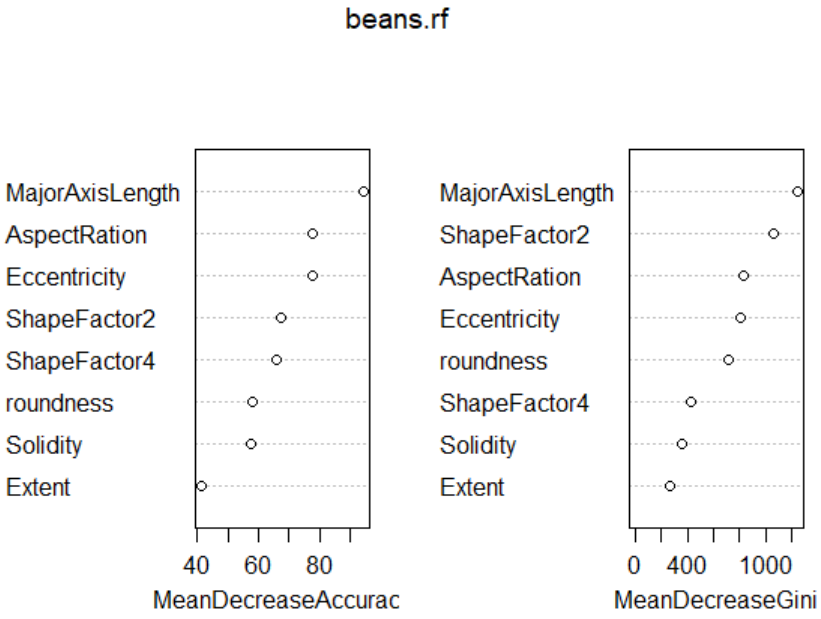

Fig. 10. Importancia de las variables para 800 árboles