Trabajo Práctico 2: Selección de variables

Minería de Datos

Alumno: Navall, Nicolás Uriel. N-1159/2.

2) DatosA

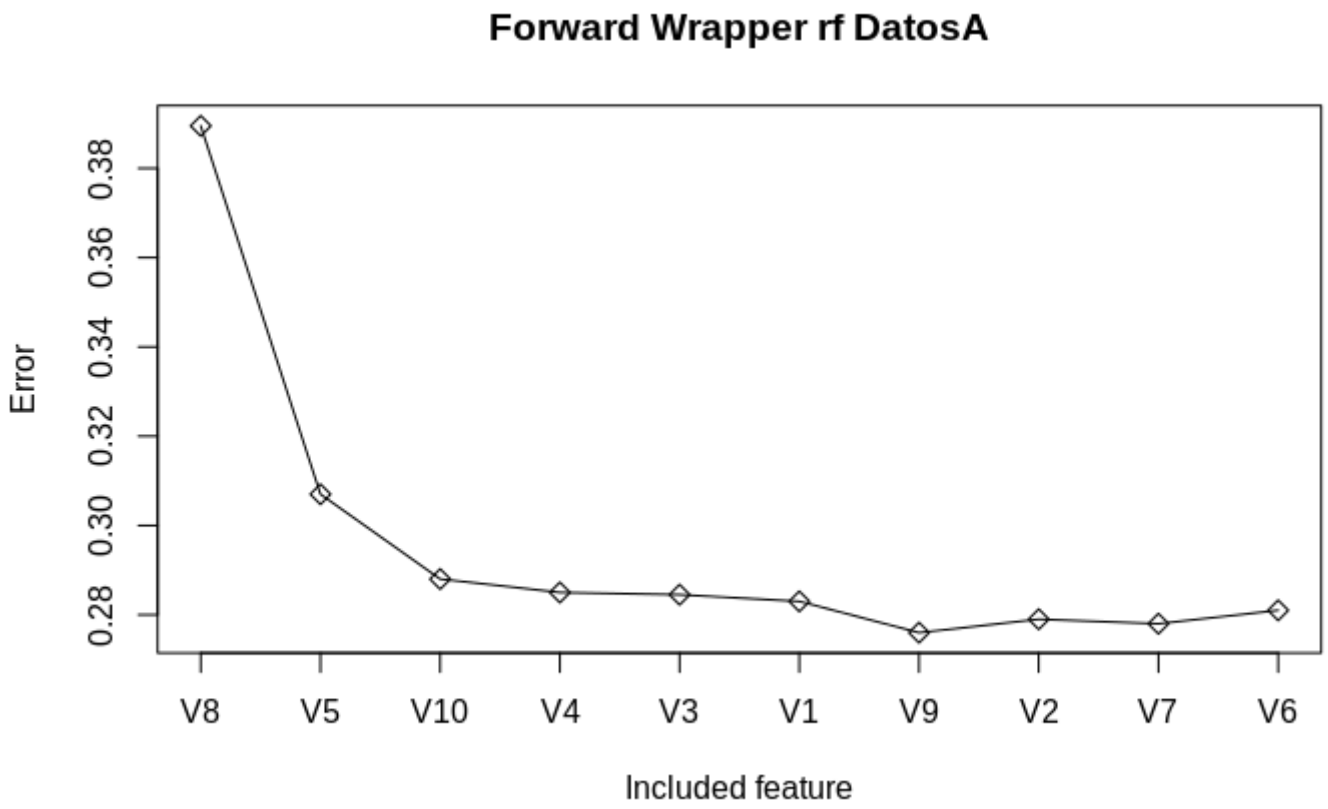

Figura 1.1

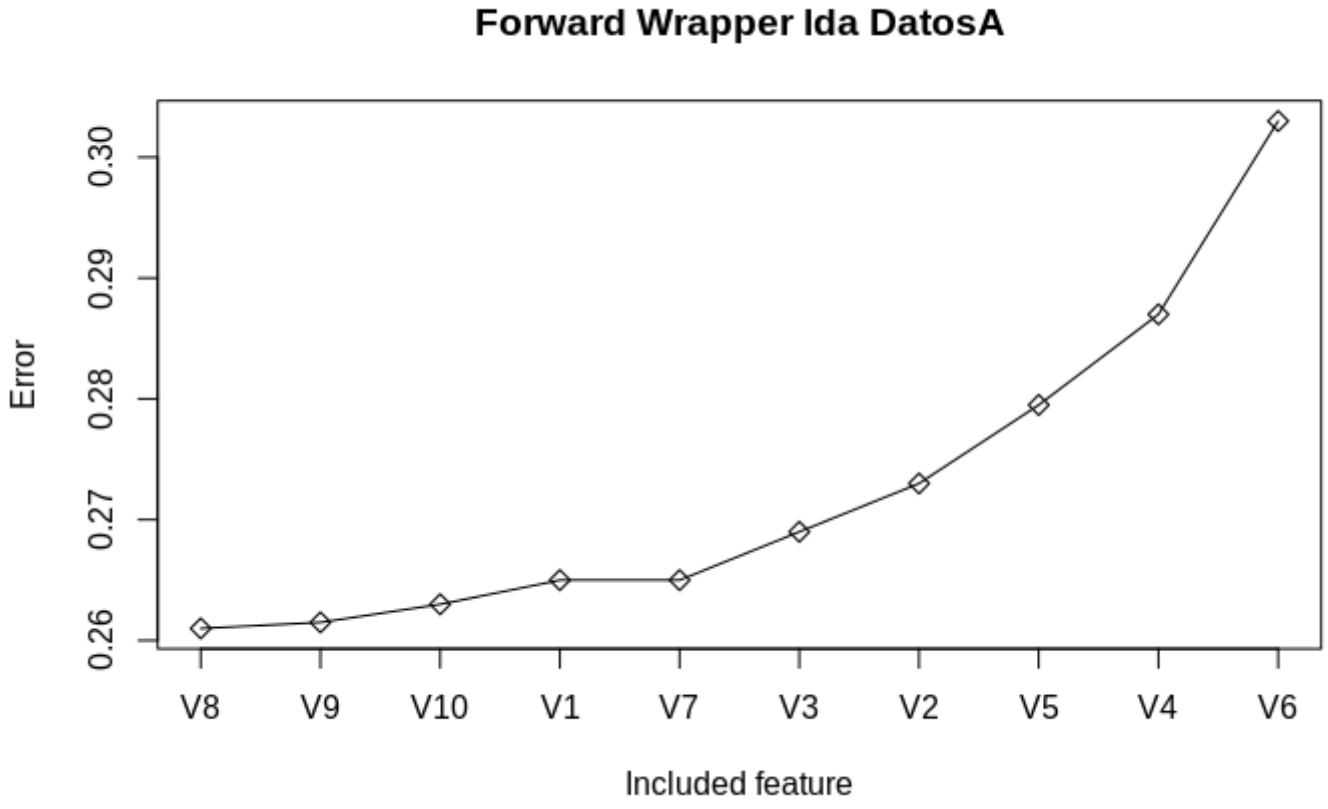

Figura 1.2

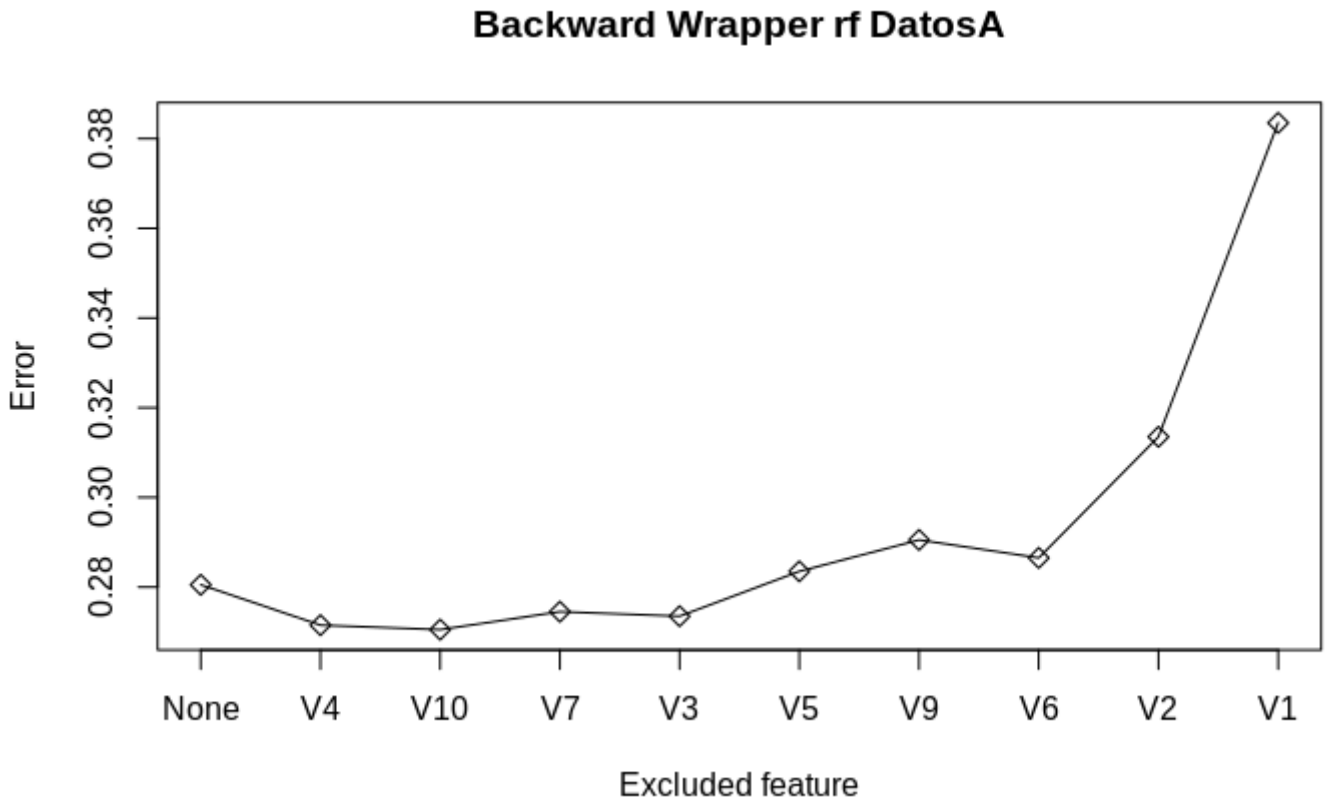

Figura 1.3

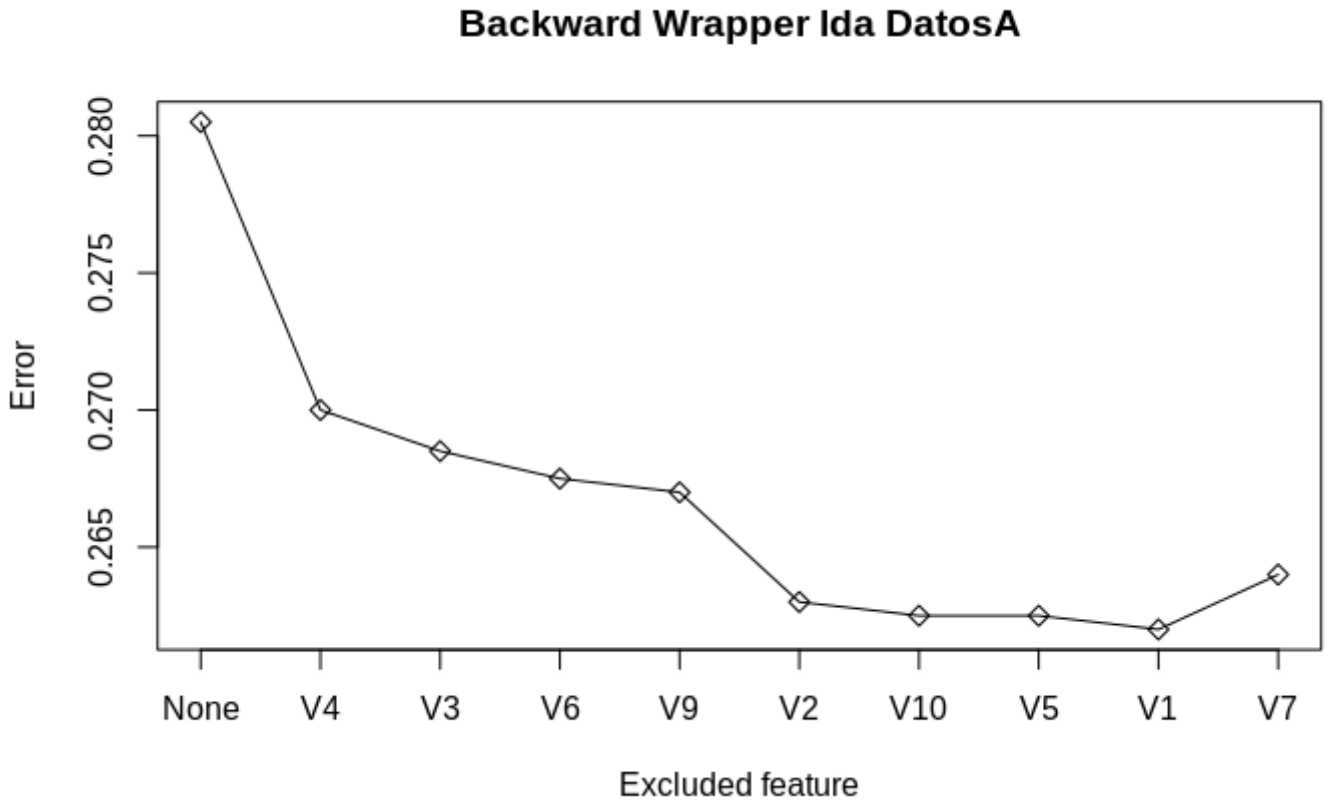

Figura 1.4

Método	Features ordenadas
Forward Wrapper rf	8 5 10 4 3 1 9 2 7 6
Forward Wrapper lda	8 9 10 1 7 3 2 5 4 6
Backward Wrapper rf	8 1 2 6 9 5 3 7 10 4
Backward Wrapper lda	8 7 1 5 10 2 9 6 3 4
Filtro no paramétrico Kruskal-Wallis	8 6 4 5 1 3 10 7 9 2
RFE rf	8 6 4 2 10 9 3 1 7 5
RFE linsvm	8 6 4 2 1 9 7 5 10 3

Como las variables no están correlacionadas podemos observar como el método de filtro es capaz de diferenciar de manera muy efectiva las features más importantes, ya que es un método univariado.

En la figura 1.1 podemos apreciar como el forward wrapper con random forest no le importa mucho agregar más variables llegado cierto punto y como en la 1.2 al agregar más features el error aumenta por el ruido introducido de las features no relevantes.

DatosB

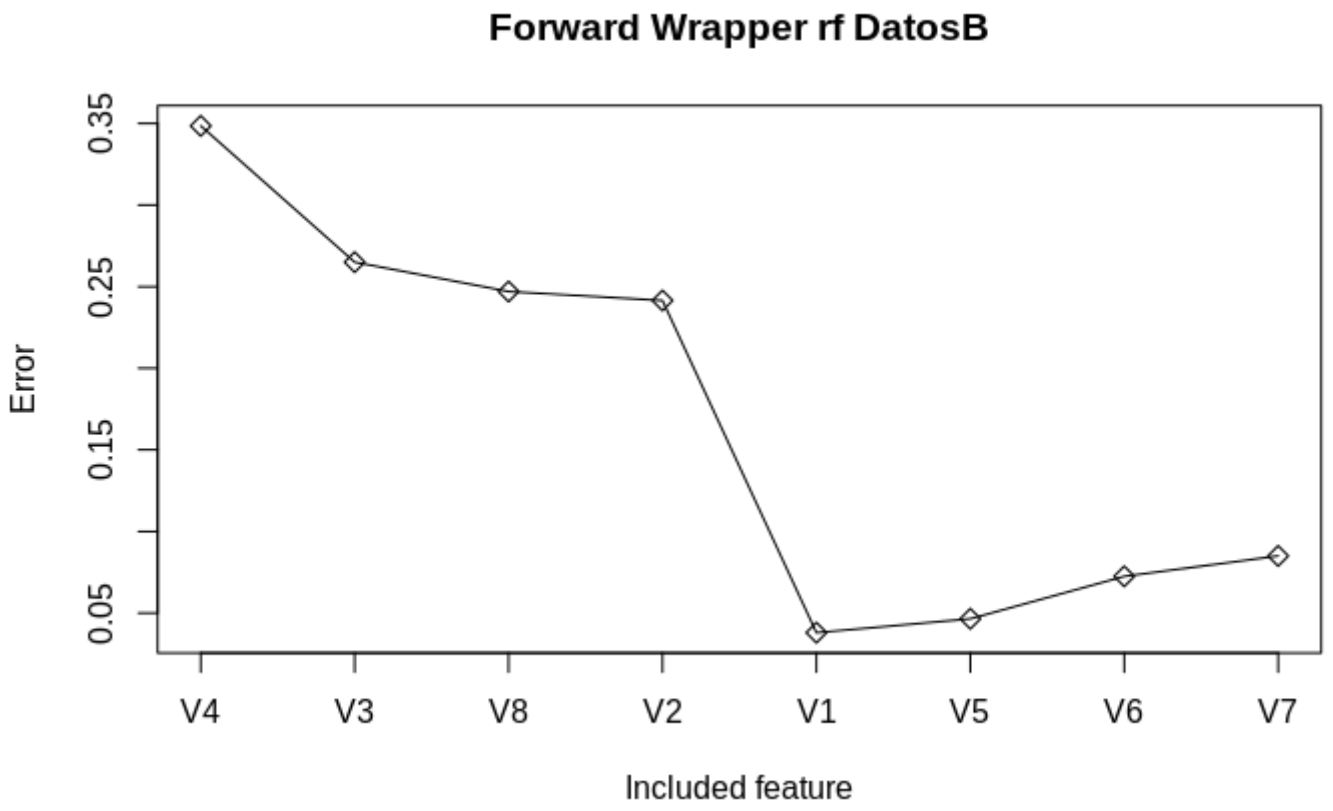

Figura 2.1

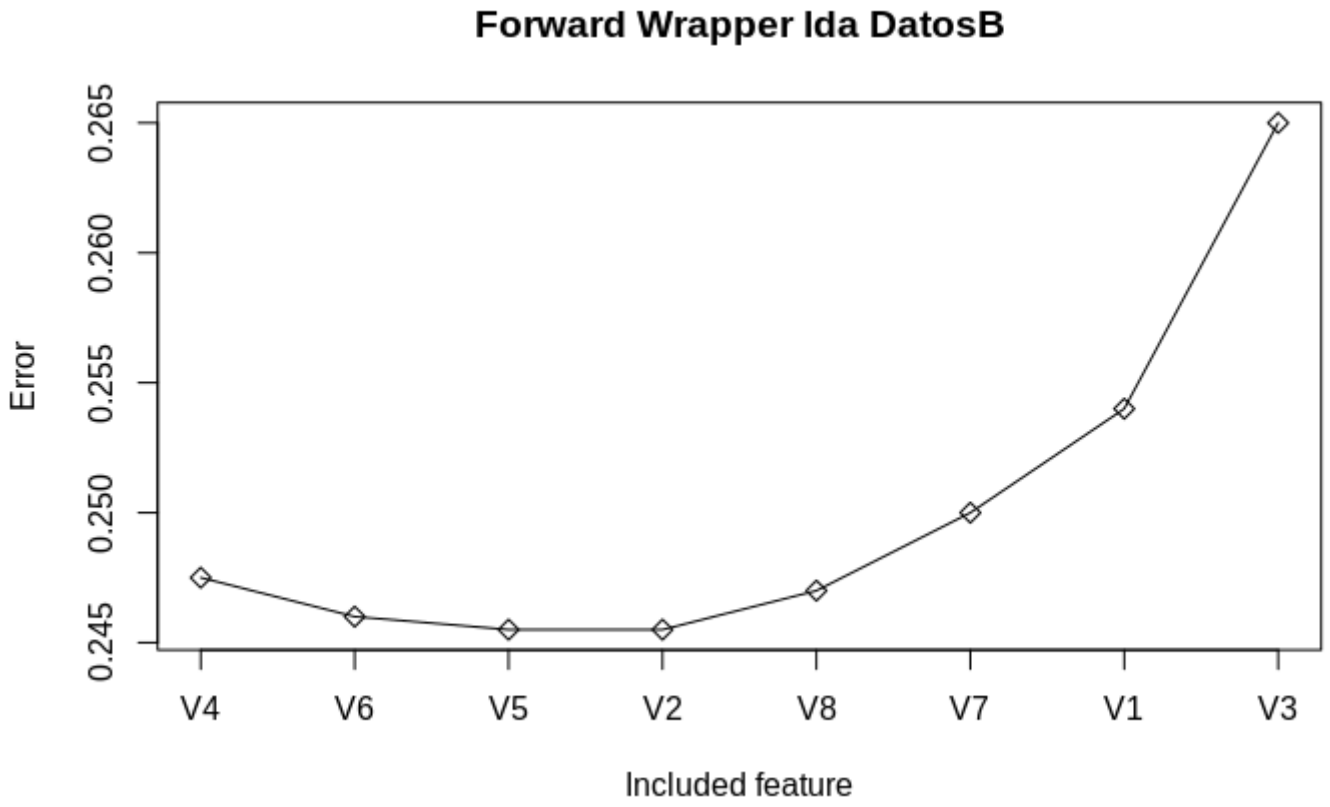

Figura 2.2

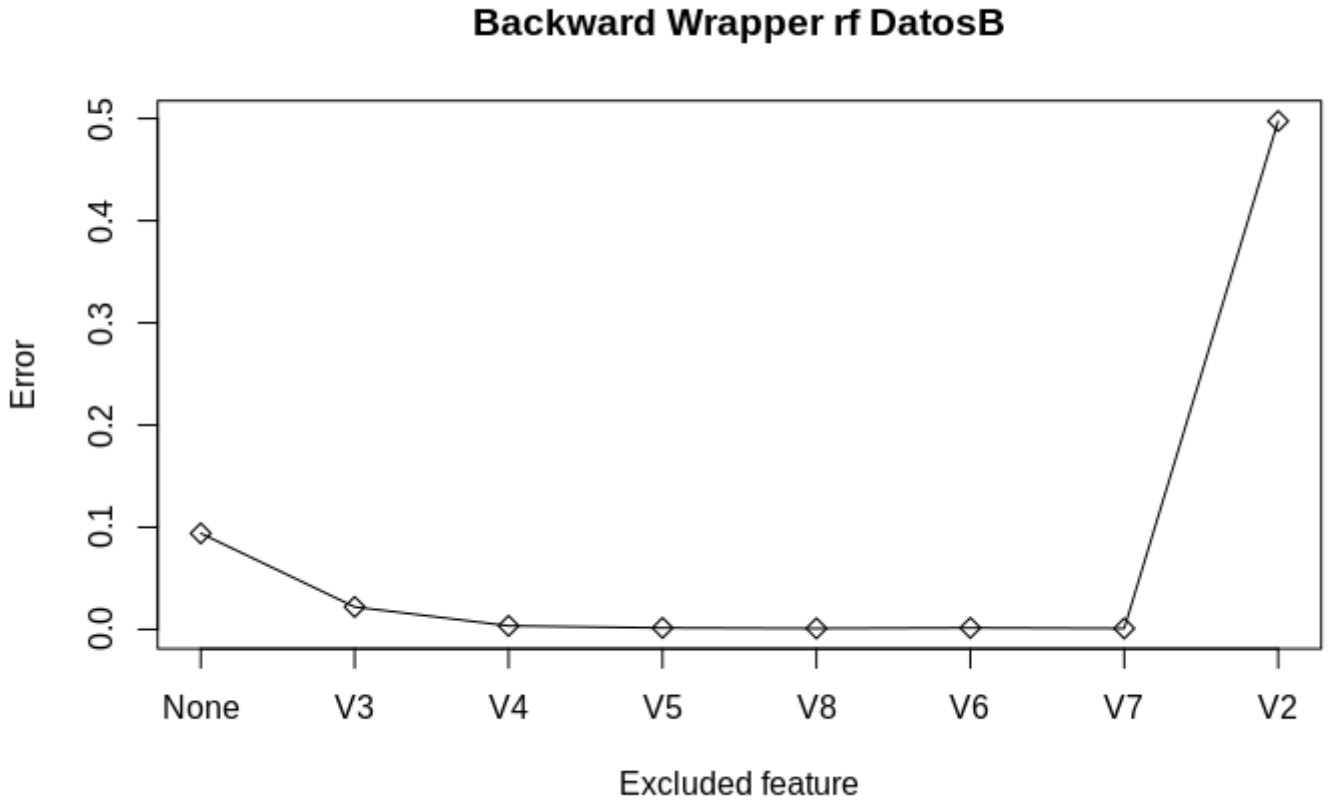

Figura 2.3

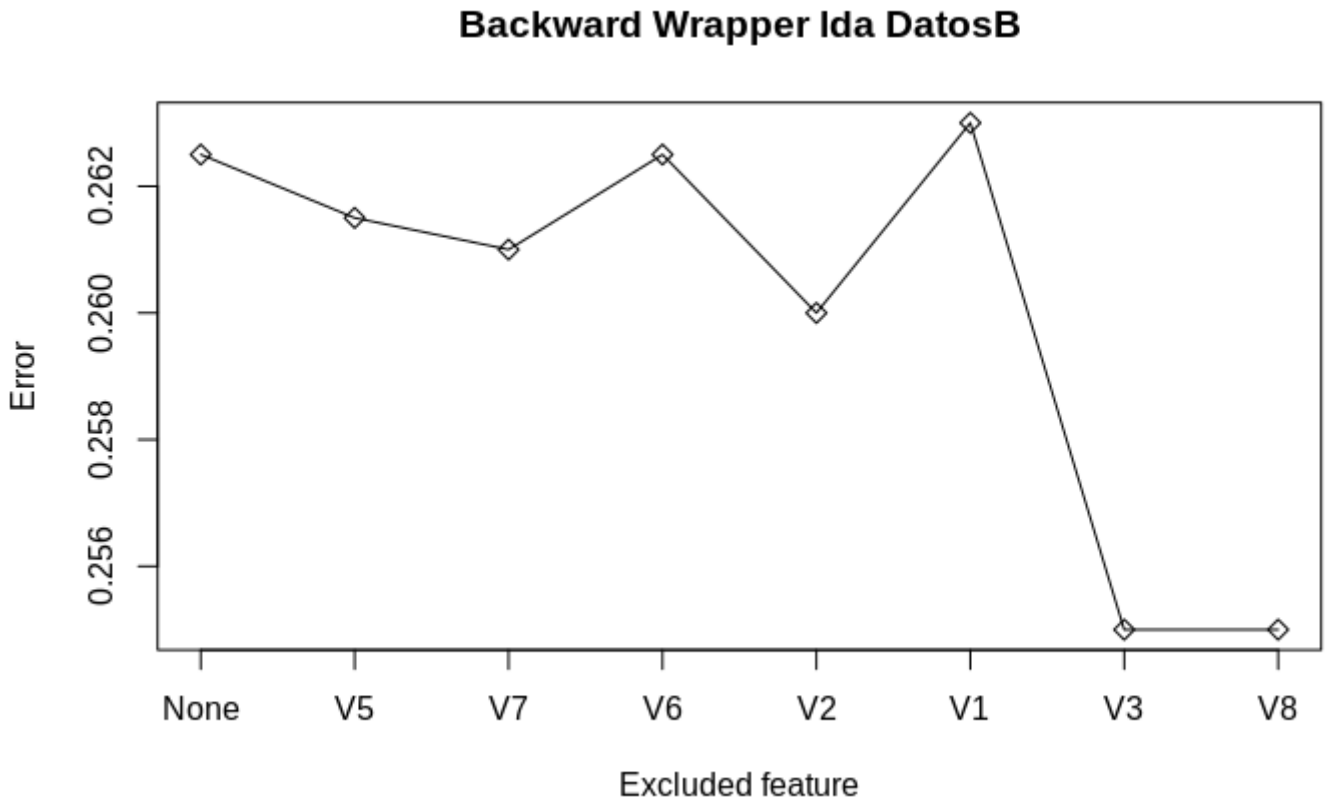

Figura 2.4

Método	Features ordenadas
Forward Wrapper rf	4 3 8 2 1 5 6 7
Forward Wrapper lda	4 6 5 2 8 7 1 3
Backward Wrapper rf	1 2 7 6 8 5 4 3
Backward Wrapper lda	4 8 3 1 2 6 7 5
Filtro no paramétrico Kruskal-Wallis	4 3 7 6 1 5 2 8
RFE rf	2 1 3 4 6 7 8 5
RFE linsvm	4 3 8 2 5 6 7 1

Como nuestros datos fueron modelados por un xor con las features 1 y 2 (y las features 3 y 4 tienen un 50% de correlación con la clase) entonces el orden ideal de las features debería tener a 1 y 2 como las más importantes seguidas de la 3 y la 4.

En el ranking dado por RFE linsvm se puede ver cómo una vez rankeada la feature 2, la feature correlacionada 1 pierde toda la importancia y pasa a estar en el último puesto del ranking.

Podemos observar en las figuras 2.1-2.4 que el error mínimo está por debajo del obtenido en las gráficas 1.1-1.4 con los datosA, lo cual tiene sentido ya que el concepto de los wrappers se basa en trabajar con subconjuntos de features, y como el problema con los datosB tiene variables correlacionadas tiene sentido que ordenar features con subconjuntos de estas de un error mínimo menor.

Veamos que el ranking dado por el filtro no es tan preciso como con los DatosA ya que las clases de los DatosB están definidas por un par de variables correlacionadas, lo cual es un problema para el método utilizado.

En la figura 2.1 podemos apreciar cómo una vez que el wrapper agrega la feature 1 ya teniendo la dos el error de este baja considerablemente, y esto se debe a que el problema está modelado de tal forma de que ambas clases son necesarias para clasificar, están correlacionadas.

3)

Método	**Promedio de features válidas en las 10 más importantes**
Forward Wrapper rf	0.1
Forward Wrapper lda	0
Backward Wrapper rf	0
Backward Wrapper lda	0.1
Filter con test no-paramétrico (Kruskal-Wallis)	0.387
RFE rf	0.35
RFE linsvm	0.267

Tiene sentido que los wrappers funcionen tan mal con los datos dados, ya que estos utilizan subconjuntos de features para decidir cuales son las más importantes, y con tantas features que actúan como ruido los wrappers deben haber hecho pasos en falso eligiendo features que corresponden al ruido o eliminando features que sirven en forward wrapper y backward wrapper respectivamente

Los RFE funcionan bien (o lo hacen en comparación de los otros métodos utilizados) por que no construye todos los modelos con una variable menos posibles, sino que rankea las variables usando una medida interna, por lo cual es menos probable que en problemas donde hay tantas features que actúan como ruido dé pasos en falso como lo haría un wrapper (al menos en un problema como este donde no tengo variables correlacionadas).

El filtro da mejor por que examina de a una variable y los datos generados no tienen variables correlacionadas, por lo que lo hace idóneo para el problema.