INTRODUCCIÓN A LA CIENCIA DE DATOS
Trabajo Teórico práctico

Alberto Castillo Lamas

Universidad de Granada
18071 Granada, España
e-mail: alcasla90@gmail.com

1. Análisis de datos

Esta sección presenta un estudio previo de cada uno de los dataset que se me han asignado.

Dataset	Instances	Attributes
Wine	178	13
autoMPG8	392	7

Table1. Características de los datasets a estudiar.

1.1. Wine

Vamos a ver un poco por encima cada una de las variables contenidas en el dataset. Primero de que tipo es cada variable, también podemos ver en la tabla 2 estadísticos básicos.

Alcohol		continua
MalicAcid		continua
Ash		continua
AlcalinityOfAsh		continua
Magnesium		discreta
TotalPhenols		continua
flavanoids	Todas son	continua
NonflavanoidsPhenols	"numeric"	continua
Proanthocyanins	(cuantitativa)	continua
ColorIntensity		continua
Hue		continua
OD280/OD315		continua
Proline		discreta
Class		3 clases

Aunque le he asignado los nombres a las variables del dataset desde el principio, debería trabajar sin ellos, no es problemma por el momento porque desconozco por completo el significado de ellos. Más adelante se estudiará el problema.

```
'data.frame':   178 obs. of  14 variables:
 $ Alcohol            : num  14.2 13.2 13.2 14.4 13.2 ...
 $ MalicAcid          : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
 $ Ash                : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
 $ AlcalinityOfAsh    : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
 $ Magnesium          : int  127 100 101 113 118 112 96 121 97 98 ...
 $ TotalPhenols       : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
 $ flavanoids         : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
 $ NonflavanoidsPhenols: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
 $ Proanthocyanins    : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
 $ ColorIntensity     : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
 $ Hue                : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
 $ OD280/OD315        : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
 $ Proline            : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
 $ Class              : int  1 1 1 1 1 1 1 1 1 1 ...
```

Figura1. Tipo y muestra de cada variable.

Todas las variables son numéricas, concretamente cuantitativas continuas, excepto una discreta. A simple vista por los datos mostrados en la tabla 2 no se aprecia nada destacable, solo algunas variables cuya mediana y media toman valores con una cierta distancia para el rango de valores de dicha variable, lo que muestra como se encuentra desbalanceada y hay una cantidad notablemente mayor de valores a un lado o a otro de la media, centrándose así la mayor parte de valores en 1 o en los dos cuartiles superiores o inferiores.

Característica	Min	Max	Median	Mean	1st Quartile	3st Quartile
Alcohol	11.03	14.83	13.05	13.00	12.36	13.68
MalicAcid	0.740	5.800	1.865	2.336	1.603	3.083
Ash	1.360	3.230	2.360	2.367	2.210	2.558
AlcalinityOfAsh	10.60	30.00	19.50	19.49	17.20	21.50
Magnesium	70.00	162.00	98.00	99.74	88.00	107.00
TotalPhenols	0.980	3.880	2.355	2.295	1.742	2.800
flavanoids	0.340	5.080	2.135	2.029	1.205	2.875
NonflavanoidsPhenols	0.1300	0.6600	0.3400	0.3619	0.2700	0.4375
Proanthocyanins	0.410	3.580	1.555	1.591	1.250	1.950
ColorIntensity	1.280	13.000	4.690	5.058	3.220	6.200
Hue	0.4800	1.7100	0.9650	0.9574	0.7825	1.1200
OD280/OD315	1.270	4.000	2.780	2.612	1.938	3.170
Proline	278.0	1680.0	673.5	746.9	500.5	985.0
Class	1.000	3.000	2.000	1.938	1.000	3.000

Table2. Características de los datasets a estudiar.

Vamos a ver algunos de estos valores representados de forma general gráficamente en la figura 2. Ahora se puede observar claramente como la variable *Proline* se mueve en un rango mucho más amplio al del resto de variables, que si se asemejan entre si (no se aprecian el resto de varibles se pueden observar en la figura 3). Por tanto para trabajar con distancias será necesario la normalizacion de los datos.

Veamos como se distribuyen los valores de cada variable, figura 4. El último gráfico muestra la distribución de la clase, la cual toma tres valores y sus frecuencias no difieren notablemente, nos nos aporta información útil. Se puede destacar también como la frecuencia de las variables *Alcohol*, *Ash*, *AlcalinityO-*

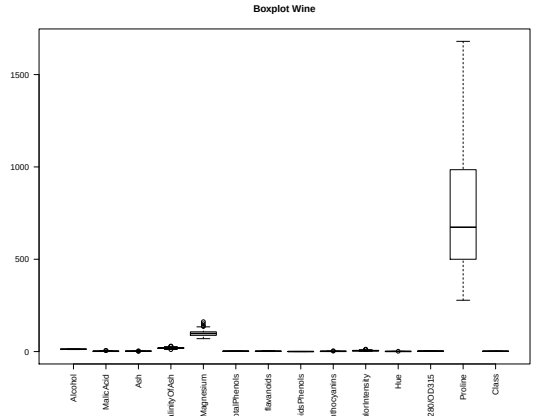

Figura2. Histograma de todas la variables del dataset Wine.

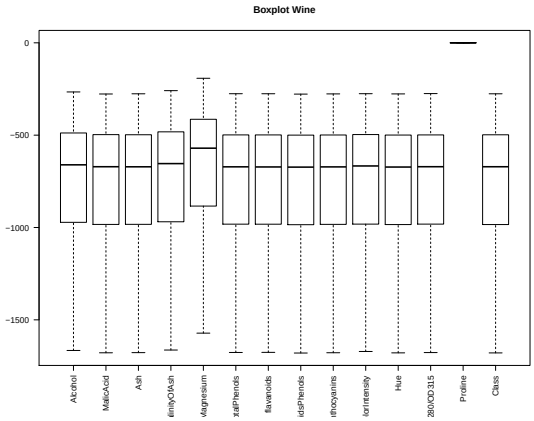

Figura3. Histograma de todas la variables del dataset Wine, menos *proline*, para poder apreciar el resto.

fAsh, *NonflavanoidsPhenols*, *Proanthocyanins* y *Hue* toman una forma similar a la de una distribución normal, podrían ser buenos estimadores.

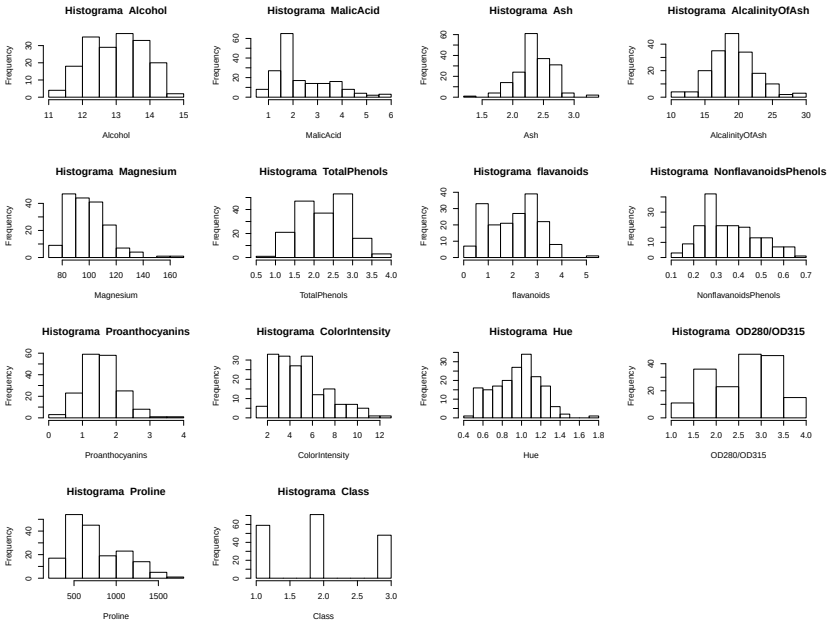

Figura4. Histograma de cada variable del dataset Wine.

Observemos en la tabla 3 como mediana, moda y media en estas variables debe ser próxima.

Así es, las variables vistas a modo gráfico cumple la anterior teoría, además se observa como la variable *TotalPhenols* aunque a simple vista no la he detectado si se observa además de por los estadístico una cierta similitud con una distribución normal, aunque quizá menos que las anteriores mencionadas que si se muestran claramente, obsérvese también la figura 5.

Por otro lado, voy a observar el resumen estadístico de cada característica respecto a la variable de clase. Se observa como este dataset no contiene ningún valor perdido en ninguna de sus 13 variables. También observamos algunos detalles como las tres clases para clasificar el vino, cuyos individuos pertenecientes a la clase **1** son 59, a las clase **2** pertenecen 71, y a la clase **3** pertenecen 48.

Característica	Median	Mean	Mode	Modal skewness
Alcohol	13.05	13.00	12.71	0.2247191
MalicAcid	1.865	2.336	1.73	0.1966292
Ash	2.360	2.367	2.29	0.247191
AlcalinityOfAsh	19.50	19.49	20	-0.1629213
Magnesium	98.00	99.74	88	0.5449438
TotalPhenols	2.355	2.295	2.2	0.1011236
flavanoids	2.135	2.029	2.65	-0.3370787
NonflavanoidsPhenols	0.3400	0.3619	0.345	-0.06741573
Proanthocyanins	1.555	1.591	1.35	0.3426966
ColorIntensity	4.690	5.058	3.666667	0.3483146
Hue	0.9650	0.9574	1.04	-0.2022472
OD280/OD315	2.780	2.612	2.87	-0.1404494
Proline	673.5	746.9	600	0.2134831
Class	2.000	1.938	2	-0.06179775

Table3. Mediana, media, moda y asimetría de la moda de cada variable del dataset Wine. Señalados en azul las variables identificadas en las gráficas como similares a una distribución normal

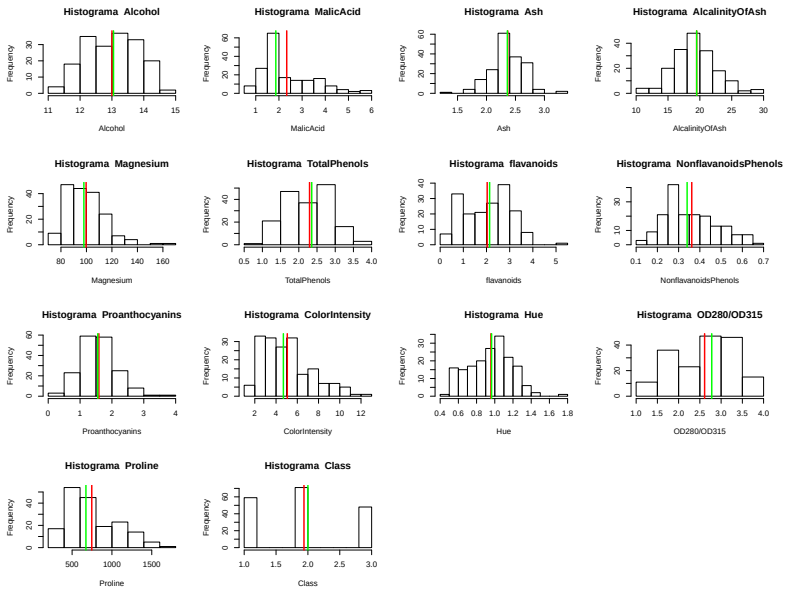

Figura5. Histograma de cada variable del dataset Wine pintando su media (línea roja) y su mediana (línea verde).

Tras estudiar cada una de las tablas se observan resultados interesantes, cómo en ciertas características y en concreto para cada una de las clase de vino los valores se agrupan entorno a un valor. En la figura 6 se puede observar un cuadro de estadísticos (obtenido para cada variable) con el que podemos comparar la media de los valores para una clase de vino y su desviación respecto a esta. Siendo así, podemos calcular el coeficiente de variación (dispersión de los valores respecto a la media) y comparar entre las características. Esto nos muestra como las características *Alcohol* y *Ash* (figura 7) contiene unos valores muy agrupados entorno a cada media de los valores asociados a cada clase de vino. En ambas variables para la clase *1* y *3* los valores difieren por debajo del 10 %.

		N	min	Q1	median	Q3	max	mean	sd	n	missing
Class	1	59	12.85	13.4000	13.750	14.1000	14.83	13.74475	0.4621254	59	0
	2	71	11.03	11.9150	12.290	12.5150	13.86	12.27873	0.5379642	71	0
	3	48	12.20	12.8050	13.165	13.5050	14.34	13.15375	0.5302413	48	0
Overall		178	11.03	12.3625	13.050	13.6775	14.83	13.00062	0.8118265	178	0

Figura6. Summary (biblioteca Mosaic) de la variable Alcohol.

		sd
Class	1	0.09250961
	2	0.14053317
	3	0.07578328
Overall		0.11592734

		sd
Class	1	0.09250961
	2	0.14053317
	3	0.07578328
Overall		0.11592734

Figura7. Coeficiente de variación (Muestra nombre de columna sd pero ha sido dividido por la media) de *Alcohol* a la izquierda y de *Ash* a la derehca

Nos indica que dichas características puede que sean importantes clasificadores. Incluso aunque tengamos los valores agrupados entorno a unos valores de medias estar dos variables son casos opuestos pues aunque en *Ash* estén agrupados entorno a sus medias, estas medias están juntas y por los individuos de las clases se solapan como puede verse en la figura 9, al contrario de lo que sucede en la variable *Alcohol* cuyas clases se encuentran separadas y diferenciadas, figura 8. Casi podría clasificar solo con esta característica.

Así, aunque por su coeficiente de variación no lo aprecia hay otras variable como *TotalPhenols* y *flavanoids* que marcan los tres grupos de forma diferenciada como puede verse en las figuras 10 y 11 respectivamente.

Remarco las tres variable, *Alcohol*, *TotalPhenols*, y *flavanoids*, que gráficamente vemos como dividen el espacio de forma más o menos clara. Seguramente más adelante nos proporcionen una buena base como variables clasificadoras.

Para finalizar calculo el coeficiente de correlación de Pearson entre cada variable. Así se mostrará la dependencia para la clase de algunas variables, o la dependencia de otras variables indirectamente a través de otra, Obsérvese las tablas 4 y 5.

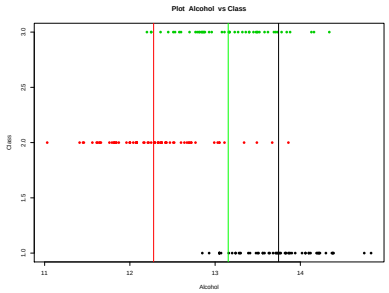

Figura8. Plot Alcohol junto con medias de cada clase.

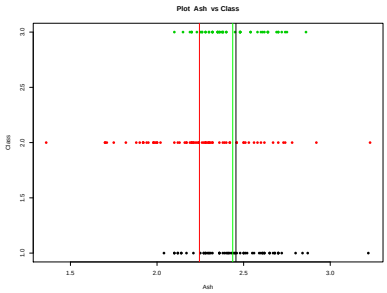

Figura9. Plot Ash junto con medias de cada clase.

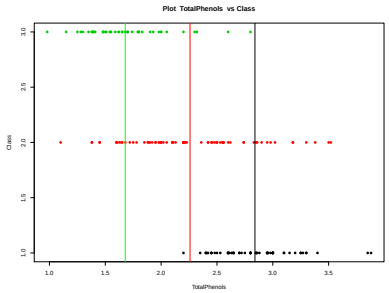

Figura10. Plot Alcohol junto con medias de cada clase.

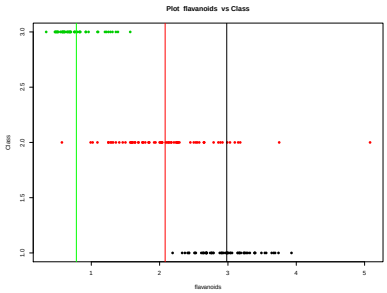

Figura11. Plot Ash junto con medias de cada clase.

Característica	Alcohol	MalicAcid	Ash	AlcalinityOfAsh	Magnesium	TotalPhenols	flavanoids
Alcohol	1.000000	0.094397	0.2115446	-0.310235	0.270798	0.289101	0.23681
MalicAcid	0.094397	1.000000	0.1640455	0.288500	-0.054575	-0.335167	-0.41101
Ash	0.211545	0.164045	1.0000000	0.443367	0.286587	0.128980	0.11508
AlcalinityOfAsh	-0.310235	0.288500	0.4433672	1.000000	-0.083333	-0.321113	-0.35137
Magnesium	0.270798	-0.054575	0.2865867	-0.083333	1.000000	0.214401	0.19578
TotalPhenols	0.289101	-0.335167	0.1289795	-0.321113	0.214401	1.000000	*0.86456
flavanoids	0.236815	-0.411007	0.1150773	-0.351370	0.195784	*0.864564	1.00000
NonflavanoidsPhenols	-0.155929	0.292977	0.1862304	0.361922	-0.256294	-0.449935	-0.53790
Proanthocyanins	0.136698	-0.220746	0.0096519	-0.197327	0.236441	*0.612413	*0.65269
ColorIntensity	0.546364	0.248985	0.2588873	0.018732	0.199950	-0.055136	-0.17238
Hue	-0.071747	-0.561296	-0.0746669	-0.273955	0.055398	0.433681	0.54348
OD280/OD315	0.072343	-0.368710	0.0039112	-0.276769	0.066004	*0.699949	*0.78719
Proline	*0.643720	-0.192011	0.2236263	-0.440597	0.393351	0.498115	0.49419
Class	-0.328222	0.437776	-0.0496432	0.517859	-0.209179	*-0.719163	*-0.84750

Table4. Coeficiente de correlación entre cada par de variables, 1°parte. Marcado con asterisco las variables con cierta correlación (¿0.6).

Característica	Nonflavan.	Proanthoc.	ColorIntens.	Hue	OD280/OD315	Proline	Class
Alcohol	-0.15593	0.1366979	0.546364	-0.071747	0.0723432	*0.64372	-0.328222
MalicAcid	0.29298	-0.2207462	0.248985	-0.561296	-0.3687104	-0.19201	0.437776
Ash	0.18623	0.0096519	0.258887	-0.074667	0.0039112	0.22363	-0.049643
AlcalinityOfAsh	0.36192	-0.1973268	0.018732	-0.273955	-0.2767685	-0.44060	0.517859
Magnesium	-0.25629	0.2364406	0.199950	0.055398	0.0660039	0.39335	-0.209179
TotalPhenols	-0.44994	*0.6124131	-0.055136	0.433681	*0.6999494	0.49811	*-0.719163
flavanoids	-0.53790	*0.6526918	-0.172379	0.543479	*0.7871939	0.49419	*-0.847498
NonflavanoidsPhenols	1.00000	-0.3658451	0.139057	-0.262640	-0.5032696	-0.31139	0.489109
Proanthocyanins	-0.36585	1.0000000	-0.025250	0.295544	0.5190671	0.33042	-0.499130
ColorIntensity	0.13906	-0.0252499	1.000000	-0.521813	-0.4288149	0.31610	0.265668
Hue	-0.26264	0.2955443	-0.521813	1.000000	0.5654683	0.23618	*-0.617369
OD280/OD315	-0.50327	0.5190671	-0.428815	0.565468	1.0000000	0.31276	*-0.788230
Proline	-0.31139	0.3304167	0.316100	0.236183	0.3127611	1.00000	*-0.633717
Class	0.48911	-0.4991298	0.265668	*-0.617369	*-0.7882296	*-0.63372	1.000000

Table5. Coeficiente de correlación entre cada par de variables, 2°parte. Marcado con asterisco las variables con cierta correlación (¿0.6).

Así podemos ver que de forma directa con la variable de clase están correladas las características *TotalPhenols*, *flavanoids*, *Hue*, *OD280/OD315*, y *Proline*. Especialmente *flavanoids* con una correlación inversa de coeficiente 0.847498.

1.2. autoMPG8

Este dataset tiene una dimensión de 392 individuos y 8 características, incluyendo la de clase. Veamos que estructura tiene estas variables en la figura 12.

Cylinders		discreta
Displacement		continua
Horse_power		discreta
Weight	Todas son	discreta
Acceleration	numéricas	continua
Model_year	(cuantitativa)	discreta
Origin		discreta
Mpg		continua

Aunque aquí si se escribe los nombres de ahora en adelante se trabaja con nombres de variables genéricos.

Nos encontramos ante 7 predictores numéricos, la mayoría expresado con decimales, lo que nos permite trabajar entre ellos con menos dificultad.

Voy a obtener una serie de estadísticos básicos para cada variable, así podremos verlo de forma general en la tabla 6. Se pueden apreciar rangos dispares entre la mayor parte de las características, lo que nos indica distancias distintas y la necesidad de procesar los datos en caso de trabajar con distancias. Además

```
'data.frame':   392 obs. of  8 variables:
 $ V1: int  4 6 4 4 6 5 4 8 8 4 ...
 $ V2: num  112 155 91 135 258 183 119 429 383 89 ...
 $ V3: int  88 107 68 84 95 77 97 198 170 71 ...
 $ V4: int  2605 2472 2025 2370 3193 3530 2300 4341 3563 1925 ...
 $ V5: num  19.6 14 18.2 13 17.8 20.1 14.7 10 10 14 ...
 $ V6: int  82 73 82 82 76 79 78 70 70 79 ...
 $ V7: int  1 1 3 1 1 2 3 1 1 2 ...
 $ V8: num  28 21 37 36 17.5 25.4 27.2 15 15 31.9 ...
```

Figura12. Estructura y muestra de cada variable del dataset AutoMPG8.

la variable *V1* tiene curiosamente el valor del tercer cuartil coincide con su máximo por lo tanto una cuarta parte de los individuos de esta características tiene como valor de V1 el máximo. También hay características como *V3*, *V5*, *V6* y *V8* cuya media y mediana son próximas lo que nos puede indicar que tienen una distribución normal, o características muy desequilibradas como *V1* o *V7*.

Característica	Min	Max	Median	Mean	1st Quartile	3st Quartile
V1	3.000	8.000	4.000	5.472	4.000	8.000
V2	68.0	455.0	151.0	194.4	105.0	275.8
V3	46.0	230.0	93.5	104.5	75.0	126.0
V4	1613	5140	2804	2978	2225	3615
V5	8.00	24.80	15.50	15.54	13.78	17.02
V6	70.00	82.00	76.00	75.98	73.00	79.00
V7	1.000	3.000	1.000	1.577	1.000	2.000
V8	9.00	46.60	22.75	23.45	17.00	29.00

Table6. Características de los datasets a estudiar.

Vemos gráficamente en la figura 13 que los rangos son dispares, sobre todo el de *V4*. Tras quitar este podemos ver en la figura 14 que hay otro grupo de dos variables, *V2* y *V3*, mocho mayores al resto. Nos queda finalmente otro grupo de variable con un rango similar, figura 15.

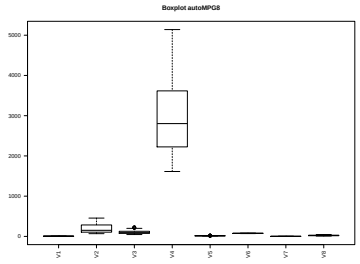

Figura13. Diagrama de cajas de cada variable del dataset AutoMPG8.

Vamos a mostrar la distribución de cada característica a través de un histograma, figura 16. Puede verse como las variable *V2*, *V3*, *V4*, *5* y *V8* siguen una distribución normal. Y aunque *V6* por su media y mediana indicaba una

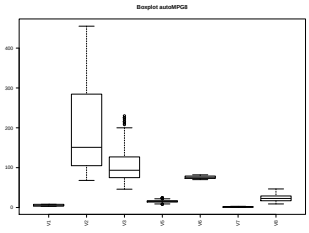

Figura14. Diagrama de cajas de cada variable del dataset AutoMPG8.

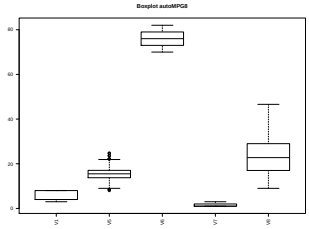

Figura15. Diagrama de cajas de cada variable del dataset AutoMPG8.

distribución normal no ha sido así. También destaco la característica *V1* cuyos individuos casi por completo se agrupan en torno a tres grupos. Esta característica si guarda cierta correlación con la variable de clase puede ser un clasificador fundamental.

Veamos junto a sus distribuciones como las variables que se han mencionado coinciden relativamente su media y media entorno al centro de su distribución, figura 17.

Voy a ver a continuación un resumen estadístico para cada característica respecto a la variable de clase dividiendo los individuos en intervalos (por ser valores continuos) según la clase.

Estudio cada tabla como la la de la figura 18, de las que se obtiene que el conjunto de datos no contiene valores perdidos en todo el conjunto. Además se puede ver a simple vista como la agrupación de los datos en torno a su rango es en casi todas las variables entorno al 20 % o más bajo, exceptuando *V6* y *V7*, por lo tanto las cinco primeras características son candidatos adecuados para clasificar por este aspecto. Destaca sobre las demás la agrupación de la característica *V3*.

Entorno a estas variables podemos trabajar para la clasificación. Con sus valores agrupados alrededor de las medias de dichos rangos podríamos encontrar alguna variable cuyos grupos se encuentren bien agrupados y distantes del resto de grupos, esto nos daría como resultado una característica importante para clasificación.

En este caso el conjunto de datos lo vamos a enfocar para regresión (sección 4.3) por tanto nos interesa más ver el comportamiento y la distribución de los

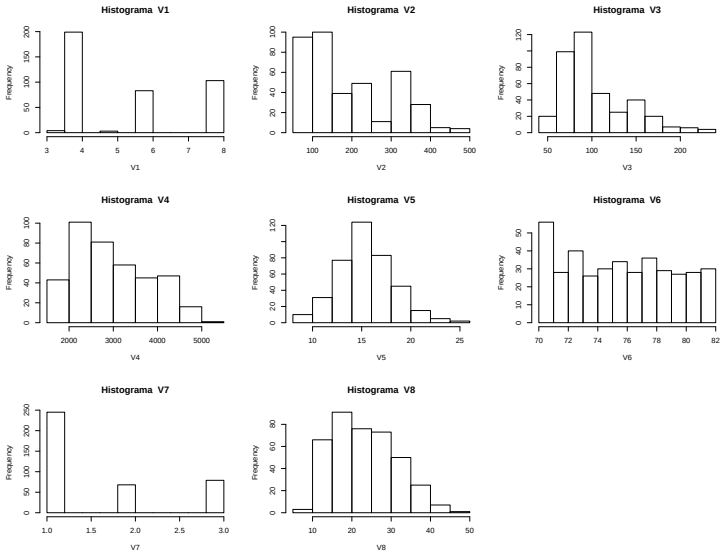

Figura16. Distribución de cada variable del dataset AutoMPG8 a través de su histograma.

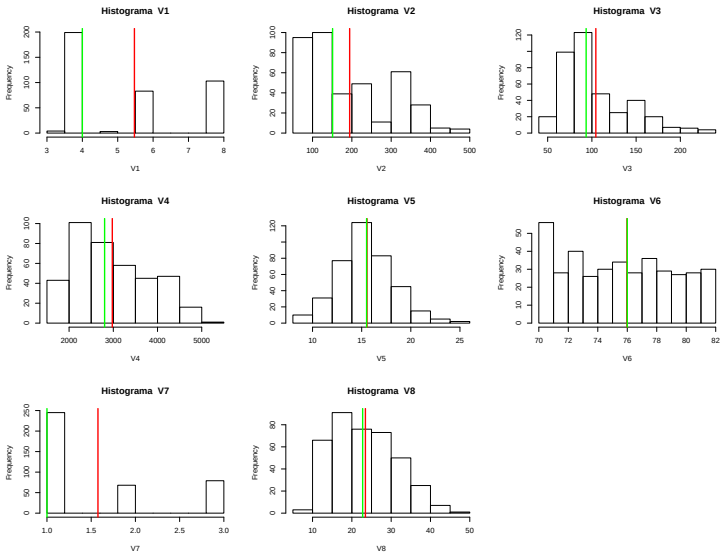

Figura17. Distribución de cada variable del dataset AutoMPG8 a través de su histograma además de pintar la media (línea roja) y mediana (línea verde).

		sd
autoMPG8Norm[, dim(autoMPG8Norm)[2]]	[0.000,0.226)	0.14008861
	[0.226,0.372)	0.26130784
	[0.372,0.545)	0.17232252
	[0.545,1.000]	0.07288642
Overall		0.34115665

		sd
autoMPG8Norm[, dim(autoMPG8Norm)[2]]	[0.000,0.226)	0.13192537
	[0.226,0.372)	0.12721028
	[0.372,0.545)	0.11178262
	[0.545,1.000]	0.08992834
Overall		0.24082862

		sd
autoMPG8Norm[, dim(autoMPG8Norm)[2]]	[0.000,0.226)	0.15998256
	[0.226,0.372)	0.16731963
	[0.372,0.545)	0.11372833
	[0.545,1.000]	0.06461036
Overall		0.27039794

		sd
autoMPG8Norm[, dim(autoMPG8Norm)[2]]	[0.000,0.226)	0.1567656
	[0.226,0.372)	0.1261090
	[0.372,0.545)	0.1491439
	[0.545,1.000]	0.1478487
Overall		0.1642181

		sd
autoMPG8Norm[, dim(autoMPG8Norm)[2]]	[0.000,0.226)	0.17999549
	[0.226,0.372)	0.10084541
	[0.372,0.545)	0.08313387
	[0.545,1.000]	0.06629718
Overall		0.20919109

Figura18. Resumen estadístico de las variables *V1* a *V5* (ordenadas de arriba a abajo y de izquierda a derecha) respecto a la variable de clase dividiendo los individuos según un rango de clase.

individuos en el espacio como muestra la figura 19. Observamos como por su distribución podemos descartar la característica *V5*. También queda en duda las variables *V1*, *V6*, y *V7*, las cuales no se ajustan de forma tan clara como *V2*, *V3* y *V4*.

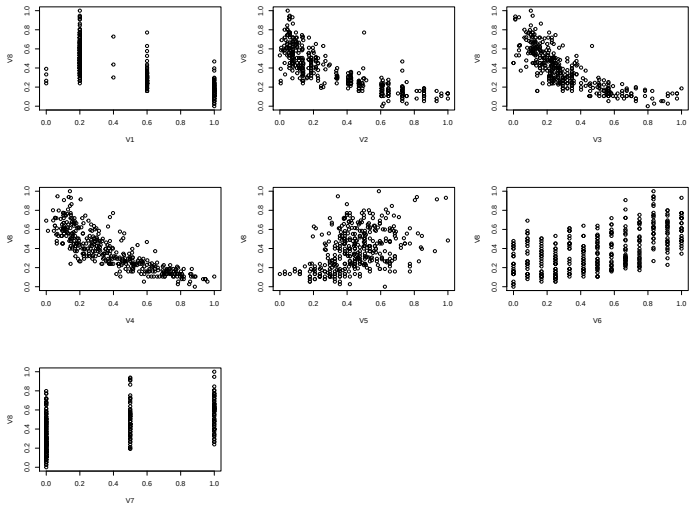

Figura19. Plot de cada variable vs la característica de clase.

Una vez vistas la distribución general de cada variable vamos a ver como esta distribución se hace particularmente para cada rango, figura 20, como muestra la tabla 7.

Entre las cinco características seleccionadas se pueden destacar *V2*, *V3* y *V4*. Muestran claramente como siguen una correlación invertida y las cuatro

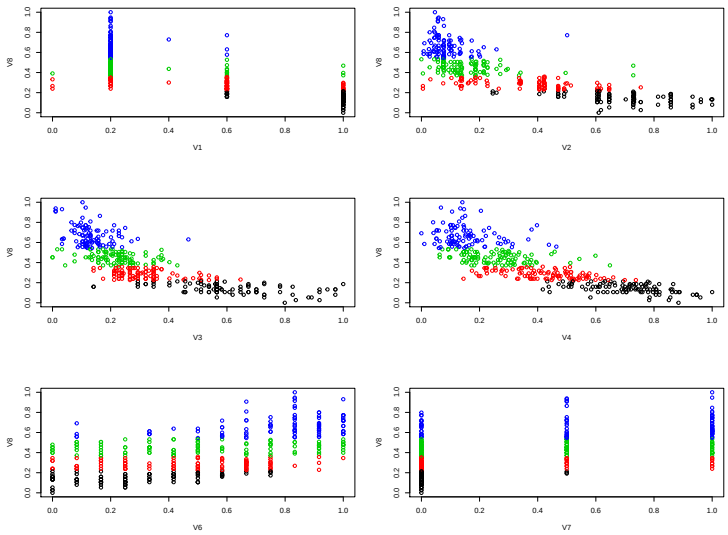

Figura20. Plot variables más interesantes con el rango de clase al que pertenecen según tabla 7.

Rango de clase	Color	Individuos
[0,000, 0,226)	Negro	99
[0,226, 0,372)	Rojo	97
[0,372, 0,545)	Verde	101
[0,545, 1,000]	Azul	95

Table7. Características de los datasets a estudiar.

clases se diferencian claramente, por tanto serán variables muy determinantes en regresión.

2. Regresión

El dataset asigando para realizar regresión es *autoMPG8*. Tiene 8 variables y por tanto 7 características regresoras. Como se explicó en la sección 1.2, podemos ver en la figura 19 como de los siete regresores podemos descartar *Acceleration* (V5) al ser una nube de puntos que no se ajusta para nada a una función lineal o curva.

Veamos ahora los indices de correlación entre las variables en la figura 8. Nos indica que las variables con mayor correlación con la clase son *Displacement*, *Horse_power*, y *Weight*. Claramente reconocible en la figura 20, pero de las tres restantes nos indica además que *Cylinders* es igual de buena que las tres anteriores, y estos cuatro regresores mantienen una correlación invertida con la clase.

Del resto de regresores no destaca ninguno, el mejor de estos es *Model_year* que completaría los 5 seleccionados para regresión, aunque yo me quedaría solo con los *Cylinders*, *Displacement*, *Horse_power*, y *Weight*.

Característica	Cylinders	Displacem.	Horse_pow.	Weight	Acceler.	Model_year	Origin	Mpg
Cylinders	1.0000	0.9508	0.8430	0.8975	-0.5047	-0.3456	-0.5689	**-0.7776
Displacement	0.9508	1.0000	0.8973	0.9330	-0.5438	-0.3699	-0.6145	**-0.8051
Horse_power	0.8430	0.8973	1.0000	0.8645	-0.6892	-0.4164	-0.4552	**-0.7784
Weight	0.8975	0.9330	0.8645	1.0000	-0.4168	-0.3091	-0.5850	**-0.8322
Acceleration	-0.5047	-0.5438	-0.6892	-0.4168	1.0000	0.2903	0.2127	0.4233
Model_year	-0.3456	-0.3699	-0.4164	-0.3091	0.2903	1.0000	0.1815	*0.5805
Origin	-0.5689	-0.6145	-0.4552	-0.5850	0.2127	0.1815	1.0000	0.5652
Mpg	-0.7776	-0.8051	-0.7784	-0.8322	0.4233	0.5805	0.5652	1.0000

Table8. Coeficiente de correlación entre cada par de variables, 1°parte. Asterisco marca variables con cierta correlación con la clase, **(muy correlado, ¿0.7) *(moderadamente correlado).

Voy a aplicar el algoritmo de regresión lineal simple sobre cada regresor seleccionado, podemos ver el resumen estadístico de cada uno en las figuras de 21 a la 25. Según el coeficiente de determinación (R^2), a excepción del regresor *Model_year* que no es bueno como dije, el resto de regresores superan el coeficiente de 0,6 marcando el máximo en 0,692 el modelo del regresor *Weight*, por tanto el mejor.

```
Call:
lm(formula = autoMPG8Norm$Mpg ~ autoMPG8Norm$Cylinders, data = autoMPG8Norm)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3788 -0.0847 -0.0168  0.0678  0.4765

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)             0.6181     0.0116    53.1   <2e-16 ***
autoMPG8Norm$Cylinders -0.4731     0.0194   -24.4   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.131 on 390 degrees of freedom
Multiple R-squared:  0.605,     Adjusted R-squared:  0.604
F-statistic:  597 on 1 and 390 DF,  p-value: <2e-16
```

Figura21. Resumen estadísticos para la regresión lineal simple del regresor *Cylinders*.

En la figura 26 se puede observar cuales son las regresiones lineales simples obtenidas para cada modelo

```
Call:
lm(formula = autoMPG8Norm$Mpg ~ autoMPG8Norm$Displacement, data = autoMPG8Norm)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3435 -0.0804 -0.0134  0.0625  0.4950

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                   0.58609    0.00977    60.0   <2e-16 ***
autoMPG8Norm$Displacement -0.61808    0.02306   -26.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.123 on 390 degrees of freedom
Multiple R-squared:  0.648,     Adjusted R-squared:  0.647
F-statistic:  719 on 1 and 390 DF,  p-value: <2e-16
```

Figura22. Resumen estadísticos para la regresión lineal simple del regresor *Displacement*.

```
Call:
lm(formula = autoMPG8Norm$Mpg ~ autoMPG8Norm$Horse_power, data = autoMPG8Norm)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3609 -0.0867 -0.0091  0.0735  0.4501

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                  0.6297     0.0120    52.5   <2e-16 ***
autoMPG8Norm$Horse_power  -0.7724     0.0315   -24.5   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.13 on 390 degrees of freedom
Multiple R-squared:  0.606,     Adjusted R-squared:  0.605
F-statistic:  600 on 1 and 390 DF,  p-value: <2e-16
```

Figura23. Resumen estadísticos para la regresión lineal simple del regresor *Horse_power*.

```
Call:
lm(formula = autoMPG8Norm$Mpg ~ autoMPG8Norm$Weight, data = autoMPG8Norm)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3184 -0.0733 -0.0089  0.0569  0.4393

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             0.6617     0.0110    60.0   <2e-16 ***
autoMPG8Norm$Weight  -0.7173     0.0242   -29.6   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.115 on 390 degrees of freedom
Multiple R-squared:  0.693,     Adjusted R-squared:  0.692
F-statistic:  879 on 1 and 390 DF,  p-value: <2e-16
```

Figura24. Resumen estadísticos para la regresión lineal simple del regresor *Weight*.

```
Call:
lm(formula = autoMPG8Norm$Mpg ~ autoMPG8Norm$Model_year, data = autoMPG8Norm)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3197 -0.1447 -0.0117  0.1323  0.4843

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             0.1886     0.0163    11.6   <2e-16 ***
autoMPG8Norm$Model_year   0.3926     0.0279    14.1   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.169 on 390 degrees of freedom
Multiple R-squared:  0.337,     Adjusted R-squared:  0.335
F-statistic:  198 on 1 and 390 DF,  p-value: <2e-16
```

Figura25. Resumen estadísticos para la regresión lineal simple del regresor *Model_year*.

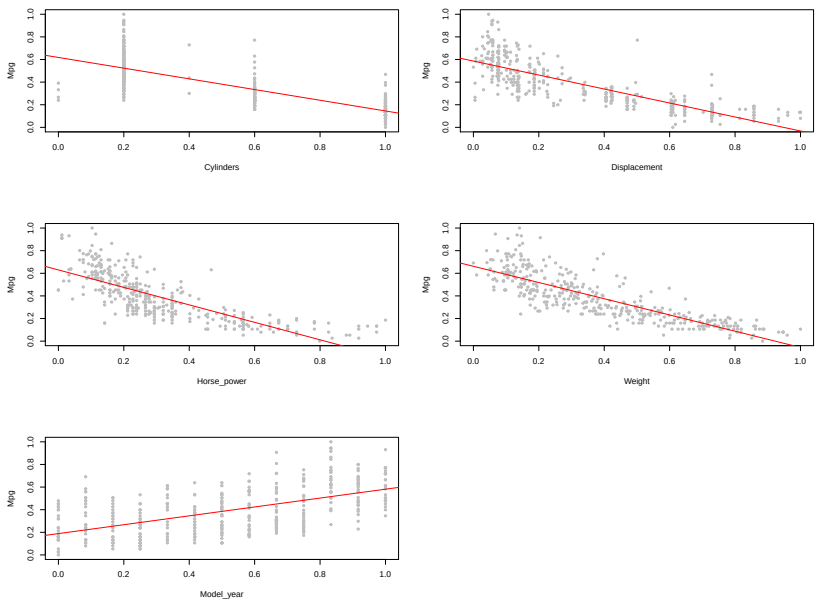

Figura26. Regresión lineal simple dibujada sobre cada plor del modelo regresor.

Una vez que hemos visto los modelos de cada regresor por separado pasamos a crear un modelo lineal múltiple teniendo en cuenta interacciones y no linealidad. Primero vamos a generar un modelo con los cinco regresores seleccionados, el cual mejora bastante el ajuste hasta un 0,806 como se ve en la figura 27. Aún así este modelo marca como variables no significativas *Cylinders*, *Desplazement*, y *Horse_power*, pero tras eliminarlas nos queda el modelo resumido por la figura 28 el cual mejora un insignificante 0,124 %.

Tengo dos modelos con con un ajuste similar pero uno de ellos está formado por solo dos regresores. Esto es positivo porque implica simplicidad en el modelo pero a la vez tiene menos margen de mejora para trabajar con no linealidad e interacción. Sigo trabajando con ambos modelos y más adelante descartaré uno de ellos.

Voy a desarrollar primero el modelo simple, será más breve. Primero aplico interacción interacción entre las variables, posteriormente añado no linealidad para cada variable (a partir de cúbico empeora), y finalmente agrego no linealidad de la interacción. El mejor modelo resultante pertenece al penúltimo paso donde se ha añadido interacción entre las dos variables y no linealidad (cuadrática) en ambas variables por separado, se muestra en la figura 29 y su ajuste es de 0,86.

Voy a desarrollar ahora el modelo con los 5 regresores. Solo añadiendo interacción entre los cuatro regresores que tenían mayor correlación ya supera al modelo de la figura 29. Este modelo muestra la importancia de *Model_year* así

```
Call:
lm(formula = Mpg ~ Cylinders + Displacement + Horse_power + Weight +
    Model_year, data = autoMPG8Norm)

Residuals:
    Min      1Q  Median      3Q     Max
-0.2359 -0.0634 -0.0024  0.0558  0.3837

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.5128     0.0154   33.22   <2e-16 ***
Cylinders     -0.0457     0.0441   -1.04     0.30
Displacement   0.0720     0.0752    0.96     0.34
Horse_power   -0.0378     0.0524   -0.72     0.47
Weight        -0.6120     0.0550  -11.12   <2e-16 ***
Model_year     0.2393     0.0167   14.30   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0913 on 386 degrees of freedom
Multiple R-squared:  0.809,     Adjusted R-squared:  0.806
F-statistic:  327 on 5 and 386 DF,  p-value: <2e-16
```

Figura27. Resumen estadístico regresión lineal múltiple con los 5 regresores seleccionados.

```
Call:
lm(formula = Mpg ~ Weight + Model_year, data = autoMPG8Norm)

Residuals:
    Min      1Q  Median      3Q     Max
-0.2354 -0.0612 -0.0031  0.0542  0.3818

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.5045     0.0135    37.4   <2e-16 ***
Weight       -0.6221     0.0201   -30.9   <2e-16 ***
Model_year    0.2417     0.0158    15.3   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0911 on 389 degrees of freedom
Multiple R-squared:  0.808,     Adjusted R-squared:  0.807
F-statistic:  819 on 2 and 389 DF,  p-value: <2e-16
```

Figura28. Resumen estadístico regresión lineal múltiple con los 5 regresores seleccionados menos los no significativos.

```
Call:
lm(formula = Mpg ~ I(Weight^2) + I(Model_year^2) + Weight * Model_year +
    Weight + Model_year, data = autoMPG8Norm)

Residuals:
    Min      1Q  Median      3Q     Max
-0.2340 -0.0455  0.0004  0.0351  0.3406

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.5870     0.0239   24.59  < 2e-16 ***
I(Weight^2)         0.6762     0.0805    8.40  8.9e-16 ***
I(Model_year^2)     0.1887     0.0503    3.75   0.0002 ***
Weight             -1.1485     0.0893  -12.86  < 2e-16 ***
Model_year          0.1317     0.0649    2.03   0.0430 *
Weight:Model_year  -0.1708     0.0738   -2.31   0.0212 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0777 on 386 degrees of freedom
Multiple R-squared:  0.862,     Adjusted R-squared:  0.86
F-statistic:  480 on 5 and 386 DF,  p-value: <2e-16
```

Figura29. Resumen estadístico regresión lineal múltiple con *Weight* y *Model_year* como regresores, añade no linealidad e interacción.

que añadiéndola y quitando de la interacción a variables menos significativas el modelo mejora aún más, ver figura 30 hasta 0,873. A continuación añado no linealidad con el que se confirma la importancia de *Model_year* como puede verse en la figura 31 dejando el **mejor modelo** con un ajuste de 0,884.

```
Call:
lm(formula = Mpg ~ Displacement * Horse_power * Weight * Model_year +
    Cylinders + Displacement + Horse_power + Weight + Model_year,
    data = autoMPG8Norm)

Residuals:
     Min       1Q   Median       3Q      Max
-0.2430 -0.0373  0.0021  0.0333  0.3179

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                                0.4880     0.0375   13.00  < 2e-16 ***
Displacement                              -0.4894     0.2299   -2.13  0.03391 *
Horse_power                                0.3352     0.2060    1.63  0.10458
Weight                                    -0.7813     0.2045   -3.82  0.00016 ***
Model_year                                 0.4655     0.0626    7.43  7.2e-13 ***
Cylinders                                  0.0702     0.0491    1.43  0.15321
Displacement:Horse_power                  -0.1974     0.3667   -0.54  0.59080
Displacement:Weight                        0.9684     0.3429    2.82  0.00500 **
Horse_power:Weight                        -0.4269     0.4063   -1.05  0.29403
Displacement:Model_year                   -0.0409     0.5161   -0.08  0.93694
Horse_power:Model_year                    -1.5042     0.4232   -3.55  0.00043 ***
Weight:Model_year                          0.2856     0.3128    0.91  0.36182
Displacement:Horse_power:Weight            0.1102     0.5182    0.21  0.83174
Displacement:Horse_power:Model_year        1.4519     1.1020    1.32  0.18846
Displacement:Weight:Model_year            -0.8817     0.8258   -1.07  0.28639
Horse_power:Weight:Model_year              0.8390     1.0501    0.80  0.42484
Displacement:Horse_power:Weight:Model_year -0.5084     1.4811   -0.34  0.73160
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.074 on 375 degrees of freedom
Multiple R-squared:  0.878,     Adjusted R-squared:  0.873
F-statistic:  169 on 16 and 375 DF,  p-value: <2e-16
```

Figura30. Resumen estadístico regresión lineal múltiple con 5 regresores, añade interacción.

```
Call:
lm(formula = Mpg ~ I(Model_year^3) + I(Model_year^2) + Displacement *
    Horse_power * Weight * Model_year + Cylinders + Displacement +
    Horse_power + Weight + Model_year, data = autoMPG8Norm)

Residuals:
     Min       1Q   Median       3Q      Max
-0.21947 -0.03692  0.00069  0.03307  0.30369

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                                0.5671     0.0382   14.84  < 2e-16 ***
I(Model_year^3)                           -0.7840     0.1735   -4.52  8.3e-06 ***
I(Model_year^2)                            1.4102     0.2705    5.21  3.1e-07 ***
Displacement                              -0.8656     0.2286   -3.79  0.00018 ***
Horse_power                                0.1726     0.1997    0.86  0.38804
Weight                                    -0.4108     0.2061   -1.99  0.04694 *
Model_year                                -0.2537     0.1368   -1.85  0.06447 .
Cylinders                                  0.0736     0.0470    1.57  0.11818
Displacement:Horse_power                   0.2866     0.3596    0.80  0.42596
Displacement:Weight                        1.1661     0.3366    3.46  0.00059 ***
Horse_power:Weight                        -0.7720     0.3928   -1.97  0.05012 .
Displacement:Model_year                    0.7026     0.5156    1.36  0.17378
Horse_power:Model_year                    -1.3027     0.4063   -3.21  0.00146 **
Weight:Model_year                         -0.1969     0.3109   -0.63  0.52701
Displacement:Horse_power:Weight           -0.1555     0.5015   -0.31  0.75671
Displacement:Horse_power:Model_year        0.4164     1.0745    0.39  0.69855
Displacement:Weight:Model_year            -1.6943     0.8266   -2.05  0.04110 *
Horse_power:Weight:Model_year              1.3670     1.0091    1.35  0.17633
Displacement:Horse_power:Weight:Model_year 0.7983     1.4631    0.55  0.58566
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0707 on 373 degrees of freedom
Multiple R-squared:  0.889,     Adjusted R-squared:  0.884
F-statistic:  166 on 18 and 373 DF,  p-value: <2e-16
```

Figura31. Resumen estadístico regresión lineal múltiple con 5 regresores, añade interacción y no linealidad.

A continuación paso a aplicar el algoritmo Knn para regresión.

Primero pruebo con todas las variables y luego utilizo el modelo obtenido en regresión lineal. Claramente mejora el resultado donde el RMSE era de 1,829 y baja hasta 1,655.

Veamos que algoritmo de regresión múltiple es mejor. Para ello voy a probar con un k-5fold del conjunto de datos AutoMPG8 aplicado sobre el algoritmo de Regresión Lineal Múltiple y KNN. En la tabla 9 se refleja como claramente KNN obtiene mejores resultados que LM.

Algoritmo	MSE train	MSE test
LM	10.79	11.4
KNN	3.552	8.107

Table9. Resulatdos MSE de train y test obtenidos para k-5fild del dataset AutoMPG8 aplicado a los algoritmos LM y KNN

Para el test de Wilcoxon tenemos unos valores de $R+ = 0$, $R- = 1$ y un p-value= 1, por tanto hay un 0 % de confianza de que sean distintos.

No puedo aplicar el test de Friedman debido a la dimensión del conjunto para su cálculo.

El test post-hoc Holm indica que no existen diferencias significativas entre ningún algoritmo, como muestra la tabla 10.

	1	2
2	1	1
3	1	-

Table10. Test post-hoc holm entre los algoritmos LM, KNN, y m5.

3. Clasificación

3.1. KNN

Vamos a clasificar según el algoritmo *k-NN* el cual se basa en distancias. Como pudimos observar en la figura 2 tenemos variables, sobre todo una respecto al resto, que tiene un rango de valores mucho mayores al resto y por tanto como se explica en la subsección 1.1 será necesario la normalizacion del dataset.

Comprobamos que tras normalizar, figura 32, los datos no han alterado su distribución inicial, figura 33, cumpliéndose para cualquier par de variables. También facilita la tarea de normalizar el hecho de que no haya valores negativos.

Los datos una vez normalizados están preparados para ser utilizados por el algoritmo *K-NN*. Pero antes de normalizar he hecho la prueba, y se puede observar en la tabla 11 la enorme diferencia en los resultados pre y pos normalizacion.

Se puede observar como la bondad de la clasificación fluctúa con el aumento del número de clusters (k), se puede ver la progresión en la figura 34. Fijándonos en la línea verde, con los datos normalizados. Se observa como el algoritmo mejora con el aumento del número de k desde un inicio de 3, hasta que alcanza la cifra de 7 para posteriormente empeorar manteniéndose en torno al 96 % de precisión en la predicción. Su máximo es obtenido para un k=6 con un acierto del 97.74510 %.

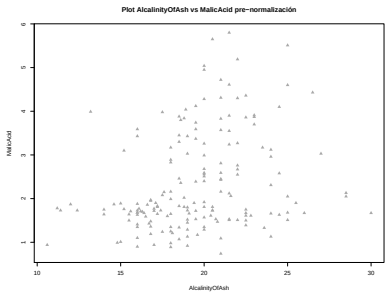

Figura32. Plot antes de normalizar datos.

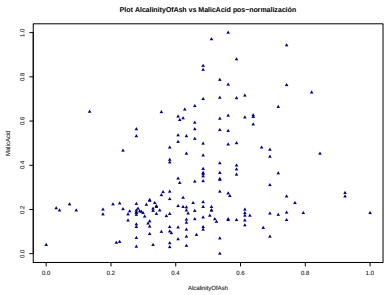

Figura33. Plot despues de normalizar datos.

k	Accuracy PreNormalizar	Accuracy PosNormalizar
3	70.29412	94.96732
4	70.71895	94.96732
5	70.68627	97.18954
6	73.00654	97.7451
7	73.03922	97.18954
8	71.92810	96.07843
9	72.48366	96.07843
10	71.33987	96.07843
11	71.30719	96.63399
12	69.60784	96.07843
13	71.30719	96.07843
14	72.97386	96.07843
15	70.75163	95.52288
16	70.22876	96.07843
17	69.08497	96.07843
18	71.33987	96.63399
19	70.75163	96.07843
20	70.78431	96.07843
21	70.22876	96.07843

Table11. Ejecuciones con distintas K para KNN antes y después de normalizar el conjunto de datos.

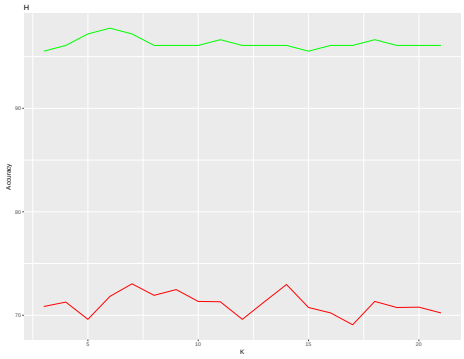

Figura34. Accuracy del algoritmo KNN con distintos números de K, pre-normalizacion (linea roja) y pos-normalización (linea verde).

3.2. LDA

En este apartado voy a realizar la predicción sobre el dataset *Wine* mediante *Linear Discriminant Analysis* o Análisis Discriminante Lineal. Mediante el cual encontraremos una combinación lineal de las características para separar en tres clases a los individuos.

Primero vamos a centrar y escalar los datos. Se puede observar en las figuras 35 (antes) y 36 (despues) la variación de algunos estadísitcos, como centrar la media.

```
    Alcohol        MalicAcid          Ash        AlcalinityOfAsh   Magnesium       TotalPhenols     flavanoids
 Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60   Min.   : 70.00   Min.   :0.980   Min.   :0.340
 1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20   1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205
 Median :13.05   Median :1.865   Median :2.360   Median :19.50   Median : 98.00   Median :2.355   Median :2.135
 Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49   Mean   : 99.74   Mean   :2.295   Mean   :2.029
 3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50   3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875
 Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00   Max.   :162.00   Max.   :3.880   Max.   :5.080
 NonflavanoidsPhenols Proanthocyanins ColorIntensity        Hue          OD280/OD315      Proline          Class
 Min.   :0.1300       Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270   Min.   : 278.0   Min.   :1.000
 1st Qu.:0.2700       1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938   1st Qu.: 500.5   1st Qu.:1.000
 Median :0.3400       Median :1.555   Median : 4.690   Median :0.9650   Median :2.780   Median : 673.5   Median :2.000
 Mean   :0.3619       Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612   Mean   : 746.9   Mean   :1.938
 3rd Qu.:0.4375       3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170   3rd Qu.: 985.0   3rd Qu.:3.000
 Max.   :0.6600       Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000   Max.   :1680.0   Max.   :3.000
```

Figura35. Summary del dataset wine sin procesar los datos.

```
    Alcohol        MalicAcid          Ash        AlcalinityOfAsh   Magnesium       TotalPhenols     flavanoids
 Min.   :-2.42739   Min.   :-1.4290   Min.   :-3.66881   Min.   :-2.663505   Min.   :-2.0824   Min.   :-2.10132   Min.   :-1.6912
 1st Qu.:-0.78603   1st Qu.:-0.6569   1st Qu.:-0.57051   1st Qu.:-0.687199   1st Qu.:-0.8221   1st Qu.:-0.88298   1st Qu.:-0.8252
 Median : 0.06083   Median :-0.4219   Median :-0.02375   Median : 0.001514   Median :-0.1219   Median : 0.09569   Median : 0.1059
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
 3rd Qu.: 0.83378   3rd Qu.: 0.6679   3rd Qu.: 0.69615   3rd Qu.: 0.600395   3rd Qu.: 0.5082   3rd Qu.: 0.80672   3rd Qu.: 0.8467
 Max.   : 2.25341   Max.   : 3.1004   Max.   : 3.14745   Max.   : 3.145637   Max.   : 4.3591   Max.   : 2.53237   Max.   : 3.0542
 NonflavanoidsPhenols Proanthocyanins ColorIntensity        Hue          OD280/OD315      Proline          Class
 Min.   :-1.8630      Min.   :-2.06321   Min.   :-1.6297   Min.   :-2.08884   Min.   :-1.8897   Min.   :-1.4890   Min.   :-1.21053
 1st Qu.:-0.7381      1st Qu.:-0.59560   1st Qu.:-0.7929   1st Qu.:-0.76540   1st Qu.:-0.9496   1st Qu.:-0.7824   1st Qu.:-1.21053
 Median :-0.1756      Median :-0.06272   Median :-0.1588   Median : 0.03303   Median : 0.2371   Median :-0.2331   Median : 0.07974
 Mean   : 0.0000      Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
 3rd Qu.: 0.6078      3rd Qu.: 0.62741   3rd Qu.: 0.4926   3rd Qu.: 0.71116   3rd Qu.: 0.7864   3rd Qu.: 0.7561   3rd Qu.: 1.37000
 Max.   : 2.3956      Max.   : 3.47527   Max.   : 3.4258   Max.   : 3.29241   Max.   : 1.9554   Max.   : 2.9631   Max.   : 1.37000
```

Figura36. Summary del dataset wine tras centrar y escalar los datos.

Una vez procesados, compruebo los predictores *near-zero variance*, figura 37. Los casos positivos serían eliminados, pero en este caso no hay ninguno.

	freqRatio	percentUnique	zeroVar	nzv
Alcohol	1.000000	70.786517	FALSE	FALSE
MalicAcid	1.750000	74.719101	FALSE	FALSE
Ash	1.000000	44.382022	FALSE	FALSE
AlcalinityOfAsh	1.363636	35.393258	FALSE	FALSE
Magnesium	1.181818	29.775281	FALSE	FALSE
TotalPhenols	1.333333	54.494382	FALSE	FALSE
flavanoids	1.333333	74.157303	FALSE	FALSE
NonflavanoidsPhenols	1.000000	21.910112	FALSE	FALSE
Proanthocyanins	1.285714	56.741573	FALSE	FALSE
ColorIntensity	1.000000	74.157303	FALSE	FALSE
Hue	1.142857	43.820225	FALSE	FALSE
OD280/OD315	1.250000	68.539326	FALSE	FALSE
Proline	1.000000	67.977528	FALSE	FALSE
Class	1.203390	1.685393	FALSE	FALSE

Figura37. Comprobamos características *near-zero variance*.

Ya podemos utilizar el algoritmo LDA para clasificar. Para ello voy a realizar varios modelos basándome en el análisis exploratorio realizado en la sección 1.1.

Voy a probar primero con las dos variables que más correlación mostraron con la clase, *flavanoids* y *OD280/OD315*. Y calculamos la calidad del ajuste comparando la predicción y las etiquetas, ver figura 38, lo que nos da un acierto del 83,70787 %. La división se puede observar gráficamente en la figura 39.

	class 1	class 2	class 3
class 1	53	14	0
class 2	6	48	0
class 3	0	9	48

Figura38. Comprobamos características *near-zero variance*.

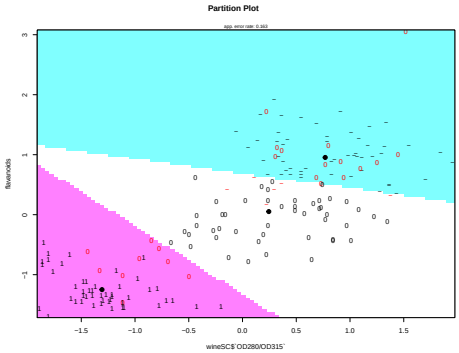

Figura39. División del espacio para cada clase según el algoritmo LDA, modelo 1.

El resultado del primer modelo es bueno pero queda lejos de la clasificación con KNN. Voy a utilizar más variables de las que estudiamos que puedan ser importantes. Utilizo *flavanoids* y *OD280/OD315* como en el modelo anterior y añadimos *Proline*, *Hue* y *TotalPhenols*, véase la figura 40.

Exactamente, el modelo mejora notablemente para subir su *accuracy* hasta 94,38202 %, colocándose algo por debajo de la precisión del algoritmo KNN.

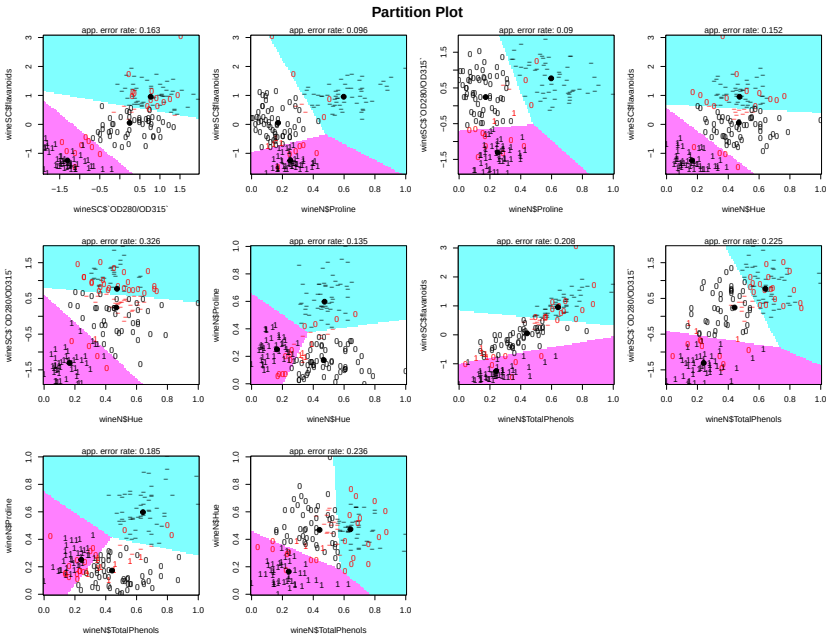

Figura40. División del espacio para cada clase según el algoritmo LDA, modelo 2.

3.3. QDA

Quadratic Discriminant Analysis funciona mejor cuando las varianzas son muy diferentes entre clases. Como podemos ver en la figura 41 hay ciertas varianzas con la misma magnitud pero aún así relativamente difieren dentro de la misma magnitud. Por tanto nos encontramos en un caso bueno donde aplicar QDA.

```
          Alcohol         MalicAcid               Ash  AlcalinityOfAsh         Magnesium      TotalPhenols
     6.590623e-01      1.248015e+00      7.526464e-02      1.115269e+01      2.039893e+02      3.916895e-01
       flavanoids NonflavanoidsPhenols      Proanthocyanins     ColorIntensity               Hue  OD280/OD315
     9.977187e-01      1.548863e-02      3.275947e-01      5.374449e+00      5.224496e-02      5.040864e-01
         Proline             Class
     9.916672e+04      6.006792e-01
```

Figura41. Varianza de las características del dataset wine.

Para comenzar voy a seguir los mismos pasos para crear los dos primeros modelos como lo hice con LDA. Primero voy a crearlo con las dos variables que creo son más influyentes como predictoras, *flavanoids* y *OD280/OD315*, dando un modelo resultante cual gráficamente particiona el espacio de la solución como

muestra la figura 42 con una precisión del 85,95506 %. Esto quiere decir que en igualdad de condiciones QDA ha mejorado en algo más de un 2 % al algoritmo LDA.

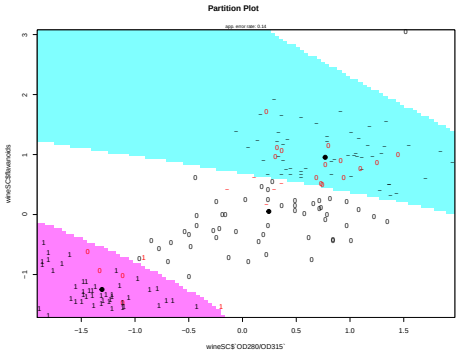

Figura42. División del espacio para cada clase según el algoritmo QDA, modelo 1.

Ahora veamos como se comporta ante el segundo conjunto de variables que apliqué a LDA. Teóricamente debería seguir mejorándolo para este conjunto de datos.

Efectivamente, sigue mejorando los resultados de LDA para situar la calidad el ajuste en un 97,19101 % lo que lo iguala con la precisión del algoritmo KNN, solo quedándose unas décimas por debajo. Su comportamiento podemos observarlo, para el conjunto de variables *flavanoids*, *OD280/OD315*, *Proline*, *Hue* y *TotalPhenols*, como muestra la figura 43.

Para finalizar, puesto que QDA ha conseguido mejorar, voy a probar a intentar mejorar el resultado obtenido con el modelo anterior a través de otra parte del estudio realizado en la sección 1.1 donde concluía con la variable *Alcohol* como una posible buena predictora debido a la distribución de sus datos. Es generado el modelo, y no puede darnos un mejor resultado mejorando la precisión de la clasificación hasta un 100 %. El modelo ha sido obtenido añadiendo al conjunto de variables del segundo modelo la nombrada característica *Alcohol*, y gráficamente puede observarse en la figura 44.

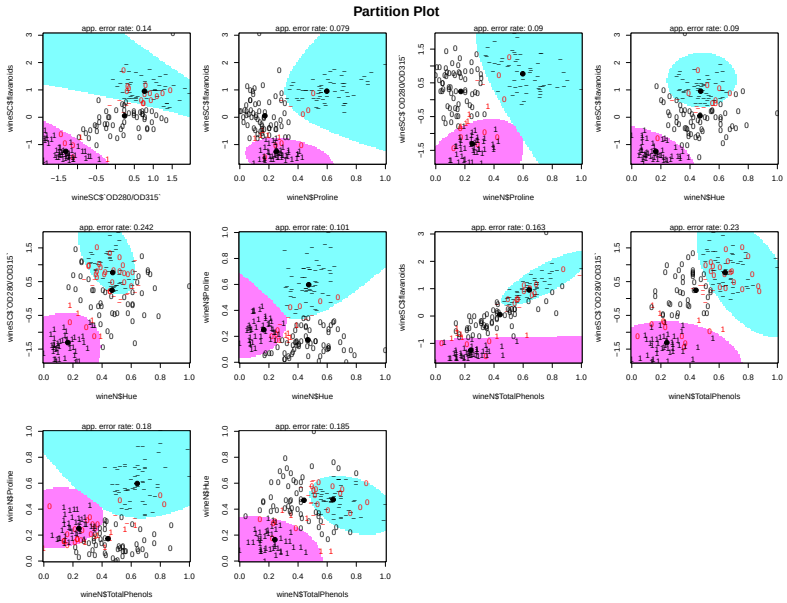

Figura43. División del espacio para cada clase según el algoritmo QDA, modelo 2.

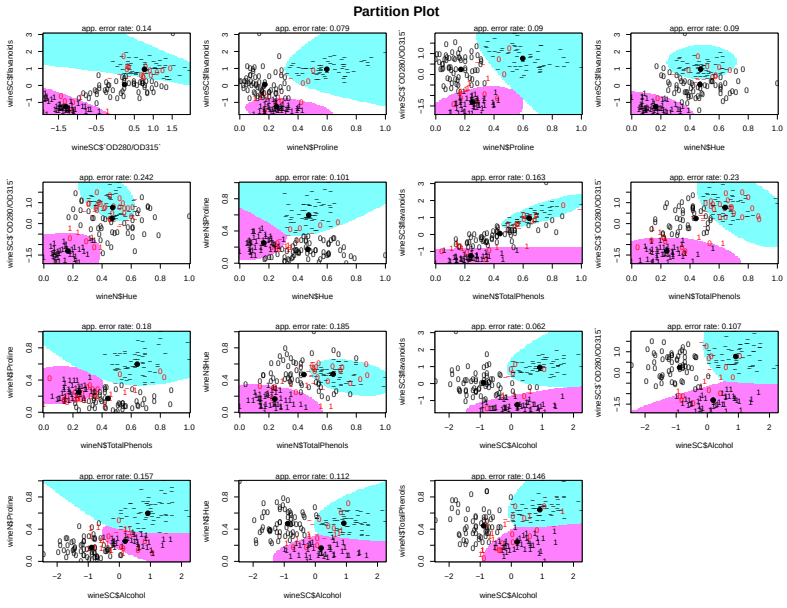

Figura44. División del espacio para cada clase según el algoritmo QDA, modelo 3.

3.4. Comparación de algoritmos

Para la comparación de los distintos algoritmos en este caso solo tenemos los resultados obtenidos para un dataset, por lo tanto no podemos hacer una ejecución por defecto de cada uno y comparar un conjunto de salidas para cada uno. En este caso voy a utilizar diferentes salidas y observar como progresa desde algo básico la calidad de la clasificación de cada uno para así poder seleccionar un ganador, no solo será basado en la precisión, se describirá brevemente el proceso para cada uno.

El algoritmo *Knn*, más estándar y aplicable a cualquier tipo de problema se ha ejecutado con los datos normalizados, necesario para su correcto funcionamiento, y probado para distintos valores de K (número de clusters) junto a todas las características del conjunto de datos. Como se puede observar en la figura 45 sin seleccionar características ya tiene una calidad en la predicción bastante alta.

Por otro lado se encuentran los algoritmos LDA y QDA, descriptivos y que nos permiten conocer la fórmula mediante la que se obtiene los resultados. En un principio, en las mismas condiciones QDA supera a LDA ligeramente, y sobre estas se impone claramente KNN (escena x=1). Pero en la evolución (escenarios x¿=2) el algoritmo KNN mejora pero no tanto como LDA y QDA, incluso en el último modelo de los tres (x=3) QDA se impone aciendo una predicción perfecta del 100 %.

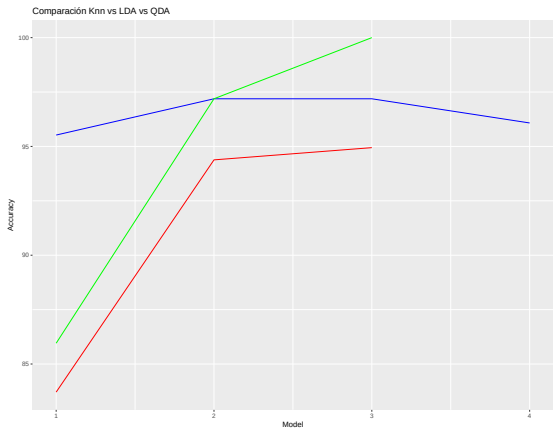

Figura45. Comparativa de ajuste de los algoritmos knn(línea azul), LDA(línea roja) y QDA(líneaa verde). En el eje X se representa los distintos escenario para cada algoritmo, siendo para Knn distintos valores de k (3,5,7,9) y para LDA y QDA modelos con distintas variables (modelo 1: OD280/OD315 + flavanoids, modelo 2: modelo 1 + Proline + Hue + TotalPhenols, modelo 3: modelo 2 + Alcohol)

Conclusión. Para este caso, concretamente con el dataset Wine, el algoritmo que mejor se comporta es QDA, no solo por la precisión en la clasificación sino también por ser descriptivo al igual que LDA. Incluso en el mejor caso KNN

podría igualar pero QDA seguiría siendo mejor opción por conocer de que forma se obtiene.

Solo en el caso de que LDA pudiera igualar a QDA tendríamos un modelo del que podemos obtener la fórmula, y además sería una fórmula más simple al ser lineal, aunque por su progresión no parece posible que LDA salve las diferencias que se imponen en igualdad de condiciones.

4. Código R

Esta sección contiene los códigos de R que permiten replicar los experimentos realizados en las secciones de análisis exploratorio, regresión y clasificación.

4.1. Análisis exploratorio de Wine

```
# Leemos dataset (***PATH AL DATASET***) y asigno nombres de cada
caracteristica
wine <- read.delim("~/R/ICC_TrabajoFinal/wine/wine.dat", header = F,
sep=",", skip=18, as.is=TRUE);
names(wine) <- c("Alcohol", "MalicAcid", "Ash", "AlcalinityOfAsh",
"Magnesium", "TotalPhenols", "flavanoids", "NonflavanoidsPhenols",
"Proanthocyanins", "ColorIntensity", "Hue", "OD280/OD315",
"Proline", "Class");
attach(wine)

dim(wine)
summary(wine)
#Obtenemos el tipo de cada variable
apply(wine, 2, mode)

# Vision general de todas la variables
boxplot(wine, main="Boxplot Wine", las=2)
boxplot(wine-wine$Proline, main="Boxplot Wine", las=2)

#Funcion que hace el histograma editando etiquetas - Dataset WINE
histograma <- function (x) {
hist(wine[,x], main=paste("Histograma ", names(wine)[x]),
xlab=names(wine)[x])
}

#Distribucion de cada variable
par(mfrow=c(4,4))
sapply(1:(dim(wine)[2]), histograma)
par(mfrow=c(1,1))

#Moda
library(modeest)
mlv(wine$Alcohol, method="mfv")
apply(wine, 2, mlv, method = "mfv")

#Histograma + media + mediana + moda
hist(wine$Alcohol, main=paste("Histograma Alcohol"), xlab="Alcohol")
abline(v=mean(wine$Alcohol), col="Red", lwd=2)
```

```
abline(v=median(wine$Alcohol), col="Green", lwd=2)

#Funcion que hace el histograma editando etiquetas, pinta su
media y mediana - Dataset WINE
histogramaAbline <- function (x) {
hist(wine[,x], main=paste("Histograma ", names(wine)[x]),
xlab=names(wine)[x])
abline(v=mean(wine[,x]), col="Red", lwd=2)
abline(v=median(wine[,x]), col="Green", lwd=2)
}

#Distribucion con media y moda
par(mfrow=c(4,4))
sapply(1:(dim(wine)[2]), histogramaAbline)
par(mfrow=c(1,1))

#Summary respecto a la clase de todas las varibales
require(Hmisc)
require(mosaic)
summaryMosaic <- function(x) {
print(names(wine)[x])
summary(wine[,x] ~ Class, data=wine, fun=favstats)
}
sapply(1:(dim(wine)[2]-1), summaryMosaic, simplify=FALSE)

#Coeficiente de variacion
coeficienteVariacion <- function(x){
print(names(wine)[x])
summary(wine[,x] ~ Class, data=wine, fun=favstats)[,8] /
summary(wine[,x] ~ Class, data=wine, fun=favstats)[,7]
}
sapply(1:(dim(wine)[2]-1), coeficienteVariacion, simplify=FALSE)

#Plot variable contra la variable de clase y pinta las medias de cada clase
para dicha variable
#Se aprecia distancias y dispersion de los datos
plotDifsMeans <- function(x) {
plot(wine[,x], wine$Class, pch=16, col=wine$Class, ylab="Class",
xlab=names(wine)[x], main=paste("Plot ",
names(wine)[x], " vs Class"))
abline(v=summary(wine[,x] ~ Class, data=wine, fun=favstats)[[4*6+1]],
col="black")
abline(v=summary(wine[,x] ~ Class, data=wine, fun=favstats)[[4*6+2]],
col="red")
abline(v=summary(wine[,x] ~ Class, data=wine, fun=favstats)[[4*6+3]],
```

```
col="green")
}
plotDifsMeans(1)
plotDifsMeans(3)
plotDifsMeans(6)
plotDifsMeans(7)
```

4.2. Análisis exploratorio de AutoMPG8

```
#Leemos dataset (***PATH AL DATASET***)
autoMPG8 <- read.delim("~/R/ICC_TrabajoFinal/autoMPG8/autoMPG8.dat",
header = F, sep=",", skip=12, as.is=TRUE);
#Nombres mas adelante
names(autoMPG8) <- c("Cylinders", "Displacement", "Horse_power",
"Weight", "Acceleration", "Model_year", "Origin", "Mpg");
attach(autoMPG8);

dim(autoMPG8)
str(autoMPG8)
summary(autoMPG8)

#Vision general de todas la variables
boxplot(autoMPG8, main="Boxplot autoMPG8", las=2)
boxplot(autoMPG8[,-c(4)], main="Boxplot autoMPG8", las=2)
boxplot(autoMPG8[,-c(2,3,4)], main="Boxplot autoMPG8", las=2)

#Funcion que hace el histograma editando etiquetas - Dataset autoMPG8
histograma <- function (x) {
hist(autoMPG8[,x], main=paste("Histograma ", names(autoMPG8)[x]),
xlab=names(autoMPG8)[x])
}

#Distribucion de cada variable
par(mfrow=c(3,3))
sapply(1:(dim(autoMPG8)[2]), histograma)
par(mfrow=c(1,1))

#Funcion que hace el histograma editando etiquetas, pinta su media
y mediana - Dataset WINE
histogramaAbline <- function (x) {
hist(autoMPG8[,x], main=paste("Histograma ", names(autoMPG8)[x]),
xlab=names(autoMPG8)[x])
abline(v=mean(autoMPG8[,x]), col="Red", lwd=2)
abline(v=median(autoMPG8[,x]), col="Green", lwd=2)
}
```

```
#Distribucion con media y moda
par(mfrow=c(3,3))
sapply(1:(dim(autoMPG8)[2]), histogramaAbline)
par(mfrow=c(1,1))


#Normalizacion
normalize <- function(x) {
(x - min(x, na.rm=TRUE))/(max(x,na.rm=TRUE) - min(x, na.rm=TRUE))
}
autoMPG8Norm <- as.data.frame(lapply(autoMPG8, normalize))


#Summary respecto a la clase de todas las varibales
require(Hmisc)
require(mosaic)
summaryMosaic <- function(x) {
print(names(autoMPG8Norm)[x])
summary(autoMPG8Norm[,x] ~ autoMPG8Norm[,dim(autoMPG8Norm)[2]],
data=autoMPG8Norm, fun=favstats)
}

sapply(1:(dim(autoMPG8Norm)[2]-1), summaryMosaic, simplify=FALSE)

#Summary respecto a la clase de todas las varibales - Muestra SD
summaryMosaic <- function(x) {
print(names(autoMPG8Norm)[x])
summary(autoMPG8Norm[,x] ~ autoMPG8Norm[,dim(autoMPG8Norm)[2]],
data=autoMPG8Norm, fun=favstats)
}

sapply(1:5, summaryMosaic, simplify=FALSE)

#Funcion que pinta del dataset 'autoMPG8' con un plot las dos clases
que se especifiquen por parametros
plotY <- function (x,y) {
plot(autoMPG8Norm[,y]~autoMPG8Norm[,x], xlab=names(autoMPG8Norm)[x],
 ylab=names(autoMPG8Norm)[y])
}

#Agrupamos graficas en rejilla (row*col) de 3x3
par(mfrow=c(3,3))
#Aplica la funcion anterior sobre todas las variables menos la clase
a predecir contra la que visualizamos el resto de variables
sapply(1:(dim(autoMPG8Norm)[2]-1), plotY, dim(autoMPG8Norm)[2])
```

```
#Restablece a la visualizacion por defecto
par(mfrow=c(1,1))


#[0.000,0.226)
#[0.226,0.372)
#[0.372,0.545)
#[0.545,1.000]
divide_class_ranges <- function(x) {
if(autoMPG8Norm$class[x]<0.226)
autoMPG8Norm$class[x]=0
else
if(autoMPG8Norm$class[x]<0.372)
autoMPG8Norm$class[x]=1
else
if(autoMPG8Norm$class[x]<0.545)
autoMPG8Norm$class[x]=2
else
autoMPG8Norm$class[x]=3
}
autoMPG8Norm$class <- autoMPG8Norm$V8
autoMPG8Norm$class <- sapply(1:dim(autoMPG8Norm)[1], divide_class_ranges)

par(mfrow=c(3,2))
plot(autoMPG8Norm$V8~autoMPG8Norm$V2, xlab=names(autoMPG8Norm)[2],
 ylab=names(autoMPG8Norm)[8], col=autoMPG8Norm$class+1)
plot(autoMPG8Norm$V8~autoMPG8Norm$V3, xlab=names(autoMPG8Norm)[3],
ylab=names(autoMPG8Norm)[8], col=autoMPG8Norm$class+1)
plot(autoMPG8Norm$V8~autoMPG8Norm$V4, xlab=names(autoMPG8Norm)[4],
ylab=names(autoMPG8Norm)[8], col=autoMPG8Norm$class+1)
plot(autoMPG8Norm$V8~autoMPG8Norm$V5, xlab=names(autoMPG8Norm)[5],
ylab=names(autoMPG8Norm)[8], col=autoMPG8Norm$class+1)
plot(autoMPG8Norm$V8~autoMPG8Norm$V6, xlab=names(autoMPG8Norm)[6],
 ylab=names(autoMPG8Norm)[8], col=autoMPG8Norm$class+1)
par(mfrow=c(1,1))
```

4.3. Regresión

```
#Leemos dataset (***PATH AL DATASET***)
autoMPG8 <- read.delim("~/R/ICC_TrabajoFinal/autoMPG8/autoMPG8.dat",
header = F, sep=",", skip=12, as.is=TRUE);
names(autoMPG8) <- c("Cylinders", "Displacement", "Horse_power",
"Weight", "Acceleration", "Model_year", "Origin", "Mpg")

#Normalizacion
normalize <- function(x) {
```

```
(x - min(x, na.rm=TRUE))/(max(x,na.rm=TRUE) - min(x, na.rm=TRUE))
}
autoMPG8Norm <- as.data.frame(lapply(autoMPG8, normalize))

cor(autoMPG8Norm)

#Ajutamos el modelo lineal
fit1=lm(autoMPG8Norm$Mpg~autoMPG8Norm$Cylinders, data=autoMPG8Norm)
fit2=lm(autoMPG8Norm$Mpg~autoMPG8Norm$Displacement, data=autoMPG8Norm)
fit3=lm(autoMPG8Norm$Mpg~autoMPG8Norm$Horse_power, data=autoMPG8Norm)
fit4=lm(autoMPG8Norm$Mpg~autoMPG8Norm$Weight, data=autoMPG8Norm)
fit5=lm(autoMPG8Norm$Mpg~autoMPG8Norm$Model_year, data=autoMPG8Norm)
#Visualizamos estadisticos basicos del modelo
summary(fit5)

#Modelos con regresion lienal simple
par(mfrow=c(3,2))
plot(Mpg~Cylinders, autoMPG8Norm, pch=19, cex=0.6, col="grey")
abline(fit1,col="red")
plot(Mpg~Displacement, autoMPG8Norm, pch=19, cex=0.6, col="grey")
abline(fit2,col="red")
plot(Mpg~Horse_power, autoMPG8Norm, pch=19, cex=0.6, col="grey")
abline(fit3,col="red")
plot(Mpg~Weight, autoMPG8Norm, pch=19, cex=0.6, col="grey")
abline(fit4,col="red")
plot(Mpg~Model_year, autoMPG8Norm, pch=19, cex=0.6, col="grey")
abline(fit5,col="red")
par(mfrow=c(1,1))

#modelo lineal multiple
fitMulti1=lm(Mpg~Cylinders+Displacement+Horse_power+Weight+Model_year,
data=autoMPG8Norm)
summary(fitMulti1)
fitMulti2=lm(Mpg~Weight+Model_year, data=autoMPG8Norm)
summary(fitMulti2)

fitMulti3=lm(Mpg~Cylinders*Displacement*Horse_power*Weight + Cylinders
+ Displacement + Horse_power + Weight + Model_year,
data=autoMPG8Norm)
summary(fitMulti3)
fitMulti3.1=lm(Mpg~ Displacement*Horse_power*Weight*Model_year + Cylinders
+ Displacement + Horse_power + Weight + Model_year,
data=autoMPG8Norm)
summary(fitMulti3.1)
fitMulti3.2=lm(Mpg~ I(Model_year^3) + I(Model_year^2)
```

```
+ Displacement*Horse_power*Weight*Model_year + Cylinders + Displacement
+ Horse_power + Weight + Model_year,
data=autoMPG8Norm)
summary(fitMulti3.2)

fitMulti4=lm(Mpg~ I(Weight^2) + I(Model_year^2) + Weight*Model_year + Weight
+ Model_year, data=autoMPG8Norm)
summary(fitMulti4)

#Knn
require(kknn)

fitknn <- kknn(Mpg~., autoMPG8, autoMPG8)
yprime = fitknn$fitted.values
sqrt(sum((autoMPG8$Mpg-yprime)^2)/length(yprime))

fitknn <- kknn(Mpg~I(Model_year^3) + I(Model_year^2)
+ Displacement*Horse_power*Weight*Model_year + Cylinders + Displacement +
Horse_power + Weight + Model_year, autoMPG8, autoMPG8)
yprime = fitknn$fitted.values
sqrt(sum((autoMPG8$Mpg-yprime)^2)/length(yprime))


#COMPARATIVA
#K-fold CROSS VALIDATION
nombre <- "~/R/ICC_TrabajoFinal/autoMPG8/autoMPG8"
run_lm_fold <- function(i, x, tt = "test") {
file <- paste(x, "-5-", i, "tra.dat", sep="")
x_tra <- read.csv(file, comment.char="@")
file <- paste(x, "-5-", i, "tst.dat", sep="")
x_tst <- read.csv(file, comment.char="@")
In <- length(names(x_tra)) - 1
names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
names(x_tra)[In+1] <- "Y"
names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
names(x_tst)[In+1] <- "Y"
if (tt == "train") {
test <- x_tra
}
else {
test <- x_tst
}
fitMulti=lm(Y~.,x_tra)
yprime=predict(fitMulti,test)
sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
```

```
}
mean(sapply(1:5,run_lm_fold,nombre,"train"))
mean(sapply(1:5,run_lm_fold,nombre,"test"))

run_knn_fold <- function(i, x, tt = "test") {
file <- paste(x, "-5-", i, "tra.dat", sep="")
x_tra <- read.csv(file, comment.char="@")
file <- paste(x, "-5-", i, "tst.dat", sep="")
x_tst <- read.csv(file, comment.char="@")
In <- length(names(x_tra)) - 1
names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
names(x_tra)[In+1] <- "Y"
names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
names(x_tst)[In+1] <- "Y"
if (tt == "train") {
test <- x_tra
}
else {
test <- x_tst
}
fitMulti=kknn(Y~.,x_tra,test)
fitMulti
yprime=fitMulti$fitted.values
sum(abs(test$Y-yprime)^2)/length(yprime) ##MSE
}
mean(sapply(1:5,run_knn_fold,nombre,"train"))
mean(sapply(1:5,run_knn_fold,nombre,"test"))


#Tabla resultados
tablatst <- data.frame(LM=11.400, KNN=8.107, m5p=8.350)
rownames(tablatst) <-"autoMPG8"

#Wilcoxon test
#Normalizamos la diferencias entre algoritmos LM y KNN
difs<-(tablatst[,1] -tablatst[,2]) / tablatst[,1]
#Aplicamos el test de Wilcoxon, calculamos primeros los valores
pertenecientes a R+ y R-
wilc_1_2 <-cbind(ifelse(difs<0, abs(difs)+0.1, 0+0.1),
ifelse(difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2) <-c(colnames(tablatst)[1], colnames(tablatst)[2])
head(wilc_1_2)

#Obtenemos valores totales
LMvsKNNtst<-wilcox.test(wilc_1_2[,1], wilc_1_2[,2],
```

```
alternative = "two.sided", paired=TRUE)
LMvsKNNtst$statistic #R+
pvalue<-LMvsKNNtst$p.value
LMvsKNNtst<-wilcox.test(wilc_1_2[,2], wilc_1_2[,1],
alternative = "two.sided", paired=TRUE)
LMvsKNNtst$statistic #R-

confianza <- (1-pvalue)*100
paste(round(confianza, 2), "%")

#Friedman test
test_friedman<-friedman.test(as.matrix(tablatst))
test_friedman

#Post-hoc Holm test
groups <-rep(1:dim(tablatst)[2], each=dim(tablatst)[1])
pairwise.wilcox.test(as.matrix(tablatst), groups,
p.adjust= "holm", paired = TRUE)
```

4.4. Clasificación

```
#Leemos dataset (***PATH AL DATASET***) y asigno nombres de cada
caracteristica
wine <- read.delim("~/R/ICC_TrabajoFinal/wine/wine.dat", header = F,
sep=",", skip=18, as.is=TRUE);
names(wine) <- c("Alcohol", "MalicAcid", "Ash", "AlcalinityOfAsh",
"Magnesium", "TotalPhenols", "flavanoids",
"NonflavanoidsPhenols", "Proanthocyanins", "ColorIntensity", "Hue",
"OD280/OD315",
"Proline", "Class");
attach(wine)

#Normalizacion
normalize <- function(x) {
(x - min(x, na.rm=TRUE))/(max(x,na.rm=TRUE) - min(x, na.rm=TRUE))
}
wineN<-as.data.frame(lapply(wine, normalize))

#Comprobamos no alteracion de datos
plot(wine$AlcalinityOfAsh, wine$MalicAcid, main="Plot AlcalinityOfAsh vs
MalicAcid pre-normalizacion", pch=17,
xlab="AlcalinityOfAsh", ylab="MalicAcid", col="darkgray")
plot(wineN$AlcalinityOfAsh, wineN$MalicAcid, main="Plot AlcalinityOfAsh vs
MalicAcid pos-normalizacion", pch=17,
xlab="AlcalinityOfAsh", ylab="MalicAcid", col="darkblue")
```

```
#Correlacion entre variables
cor(wine)


############## KNN ####################
library(class)

#K-fold CROSS VALIDATION automatico para WINE (normaliza los conjuntos
de train y test)
run_knn_fold <- function(i, kn) {
path="~/R/ICC_TrabajoFinal/wine/"
file <- paste(path, "wine-10-", i, "tra.dat", sep="")
x_tra <- read.csv(file, comment.char="@", header=FALSE)
x_tra<-as.data.frame(lapply(x_tra, normalize))
file <- paste(path, "wine-10-", i, "tst.dat", sep="")
x_tst <- read.csv(file, comment.char="@", header=FALSE)
x_tst<-as.data.frame(lapply(x_tst, normalize))
In <- length(names(x_tra)) - 1
names(x_tra)[1:In] <- paste ("X", 1:In, sep="")
names(x_tra)[In+1] <- "Y"
names(x_tst)[1:In] <- paste ("X", 1:In, sep="")
names(x_tst)[In+1] <- "Y"

fitKNN = knn(train=x_tra[,-14], test=x_tst[,-14], cl=x_tra$Y, k=kn)
accuracy = sum(table(fitKNN, x_tst[,14])[c(1,5,9)])*100 / nrow(x_tst)
}
mean(sapply(1:10,run_knn_fold,3))

accuracyKnn <-data.frame(k=c(3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21),
pre=c(70.84967, 71.27451, 69.60784, 71.83007, 73.03922, 71.9281, 72.48366,
71.33987, 71.30719,
69.60784, 71.30719, 72.97386, 70.75163, 70.22876, 69.08497, 71.33987,
70.75163, 70.78431, 70.22876),
pos=c(95.52288, 96.07843, 97.18954, 97.7451, 97.18954, 96.07843, 96.07843,
96.07843, 96.63399, 96.07843,
96.07843, 96.07843, 95.52288, 96.07843, 96.07843, 96.63399, 96.07843,
96.07843, 96.07843))

plot(accuracyKnn$k, accuracyKnn$pre, ylim=c(65,100))
plot(accuracyKnn$k, accuracyKnn$pos, ylim=c(65,100))

require(ggplot2)
ggplot(accuracyKnn, aes(x=accuracyKnn$k)) +
geom_line(aes(y=accuracyKnn$pre), colour="red") +
geom_line(aes(y=accuracyKnn$pos), colour="green") +
```

```
ylab("Accuracy") + xlab("K") + labs(title="H")


############## LDA ###################
library(MASS)
library(klaR)
library(caret)

attach(wineSC)

#Escalamos y centramos los datos
wineSC = as.data.frame(scale(wine))
summary(wineSC)
#Comprobamos y eliminamos caracteristicas nzv (near zero variance)
nearZeroVar(wineSC, saveMetrics = T)

#Modelo 1
lda.fit = lda(wineSC$Class ~ wineSC$flavanoids + wineN$OD280.OD315, data=wineSC)
plot(lda.fit, type="p")

lda.pred = predict(lda.fit, wineSC)

resultsLDA = table(lda.pred$class, wineSC$Class)
sum(table(lda.pred$class, wineSC$Class)[c(1,5,9)])*100 / nrow(wineSC)

partimat(Class~flavanoids+wineSC$`OD280/OD315`, data=wineSC ,method="lda")

#Modelo 2
lda.fit2 = lda(wineSC$Class ~ wineSC$flavanoids + wineSC$`OD280/OD315` +
wineSC$Proline + wineSC$Hue + wineSC$TotalPhenols, data=wineSC)
plot(lda.fit, type="both")

lda.pred2 = predict(lda.fit2, wineSC)

resultsLDA = table(lda.pred2$class, wineSC$Class)
sum(table(lda.pred2$class, wineSC$Class)[c(1,5,9)])*100 / nrow(wineSC)

partimat(wineSC$Class~wineSC$flavanoids+wineSC$`OD280/OD315`+wineSC$Proline+
wineSC$Hue+wineSC$TotalPhenols, data=wineSC ,method="lda")

#Modelo 3
lda.fit3 = lda(wineSC$Class ~ wineSC$flavanoids + wineSC$`OD280/OD315` +
wineSC$Proline + wineSC$Hue + wineSC$TotalPhenols + wineSC$Alcohol, data=wineSC)
plot(lda.fit, type="both")
```

```
lda.pred3 = predict(lda.fit3, wineSC)

resultsLDA = table(lda.pred3$class, wineSC$Class)
sum(table(lda.pred3$class, wineSC$Class)[c(1,5,9)])*100 / nrow(wineSC)

partimat(wineSC$Class~wineSC$flavanoids+wineSC$`OD280/OD315`+
wineSC$Proline+wineSC$Hue+
wineSC$TotalPhenols+wineSC$Alcohol, data=wineSC ,method="lda")

############## QDA ###################
apply(wine,2,var)

#Modelo 1
qda.fit=qda(wineSC$Class~wineSC$flavanoids+wineSC$`OD280/OD315`, data=wineSC)
plot(qda.fit, type="both")

qda.pred=predict(qda.fit,wineSC)

resultsQDA = table(qda.pred$class, wineSC$Class)
sum(table(qda.pred$class, wineSC$Class)[c(1,5,9)])*100 / nrow(wineSC)

partimat(wineSC$Class~wineSC$flavanoids+wineSC$`OD280/OD315`, data=wineSC ,
method="qda")

#Modelo 2
qda.fit2=qda(wineSC$Class~wineSC$flavanoids+wineSC$`OD280/OD315`+
wineSC$Proline+
wineSC$Hue+wineSC$TotalPhenols, data=wineSC)
plot(qda.fit2, type="both")

qda.pred2=predict(qda.fit2,wineSC)

resultsQDA = table(qda.pred2$class, wineSC$Class)
sum(table(qda.pred2$class, wineSC$Class)[c(1,5,9)])*100 / nrow(wineSC)

partimat(wineSC$Class~wineSC$flavanoids+wineSC$`OD280/OD315`+
wineN$Proline+wineN$Hue+
wineN$TotalPhenols, data=wineSC ,method="qda")

#Modelo 3
qda.fit3=qda(wineSC$Class~wineSC$flavanoids+wineSC$`OD280/OD315`+
wineSC$Proline+wineSC$Hue+wineSC$TotalPhenols+wineSC$Alcohol, data=wineSC)
plot(qda.fit2, type="both")

qda.pred3=predict(qda.fit3,wineSC)
```

```
resultsQDA = table(qda.pred2$class, wineSC$Class)
sum(table(qda.pred3$class, wineSC$Class)[c(1,5,9)])*100 / nrow(wineSC)

partimat(wineSC$Class~wineSC$flavanoids+wineSC$`OD280/OD315`+
wineN$Proline+wineN$Hue+wineN$TotalPhenols+wineSC$Alcohol,
data=wineSC ,method="qda", nplots.vert=4)


############## COMPARACION ###################
6accuracies <-data.frame( knn=c(95.52288, 97.18954, 97.18954, 96.07843),
lda=c(83.70787, 94.38202, 94.94382, NA),
qda=c(85.95506, 97.19101, 100, NA))

require(ggplot2)
ggplot(accuracies, aes(x=c(1,2,3,4))) +
geom_line(aes(y=accuracies$knn), colour="blue") +
geom_line(aes(y=accuracies$lda), colour="red") +
geom_line(aes(y=accuracies$qda), colour="green") +
ylab("Accuracy") + xlab("Model") +
labs(title="Comparacion Knn vs LDA vs QDA")
```