Predicting divorce

Treball d'Aprenentatge Automàtic I

Fernando Gastón & Marc Gàllego

22/06/2020

Introducció

L'objectiu d'aquest treball és predir si una parella es divorciarà. Les dades que farem servir per predir-ho provenen d'una enquesta que consta de 54 preguntes que s'han de respondre en l'escala de Likert (de molt en desacord a molt d'acord). D'altra banda, volem determinar quins dels factors analitzats al qüestionari són més rellevants a l'hora de què una relació de parella sigui exitosa.

En aquest treball es faran servir les eines apreses durant el curs d'Aprenentatge Automàtic. Primer durem a terme un anàlisi exploratori del conjunt de dades, analitzant les característiques de les variables i com s'ha fet la presa de dades. Després, proposarem diferents models de classificació i els validarem estimant-ne l'error de test fent servir mètodes de resampling. Finalment, escollirem el millor model i analitzarem els resultats trobats.

El codi utilitzat es pot trobar al fitxer `codi.R` adjunt i el seu ús per replicar els resultats està explicat en el fitxer `instruccions.txt`.

Estudi previ

Les dades tenen la següent estructura:

Atr1	Atr2	Atr3	Atr4	Atr5	Atr6	Atr7	Atr49	Atr50	Atr51	Atr52	Atr53	Atr54	Class
2	2	4	1	0	0	0	3	3	2	3	2	1	1
4	4	4	4	4	0	0	4	4	4	4	2	2	1
2	2	2	2	1	3	2	1	1	1	2	2	2	1
3	2	3	2	3	3	3	3	3	3	2	2	2	1
2	2	1	1	1	1	0	3	2	2	2	1	0	1
0	0	1	0	0	2	0	2	1	1	1	2	0	1

Disposem de 54 variables predictores (categòriques) i 170 observacions.

Com ja hem comentat, tant les variables predictores com la target són categòriques, així que les codificarem com a tals.

Aquest qüestionari consta d'afirmacions sobre la relació que s'han d'avaluar en una escala discreta entre 0 i 4. Alguns exemples són:

1. If one of us apologizes when our discussion deteriorates, the discussion ends.
2. I know we can ignore our differences, even if things get hard sometimes.

3. When we need it, we can take our discussions with my spouse from the beginning and correct it.

Cal tenir en compte que les dades provenen de matrimonis turcs i, malgrat ser recents, els matrimonis concertats encara representen bona part del total. En qualsevol cas, el nostre dataset no inclou cap variable més enllà de les respostes al test i la classe a predir. Tot i això, si coneixem que a l'enquesta també van obtenir informació addicional sobre les parelles enquestades: edats, fills, nivell educatiu, ingressos mensuals... No tenim aquesta informació i, per tant, no la podem fer servir com a predictora. Tot i així, és interessant i es pot trobar informació addicional a l'article original d'on hem obtingut les dades (primera referència).

Analitzem ara com està distribuïda la variable target:

Divorciats	Casats
86	84

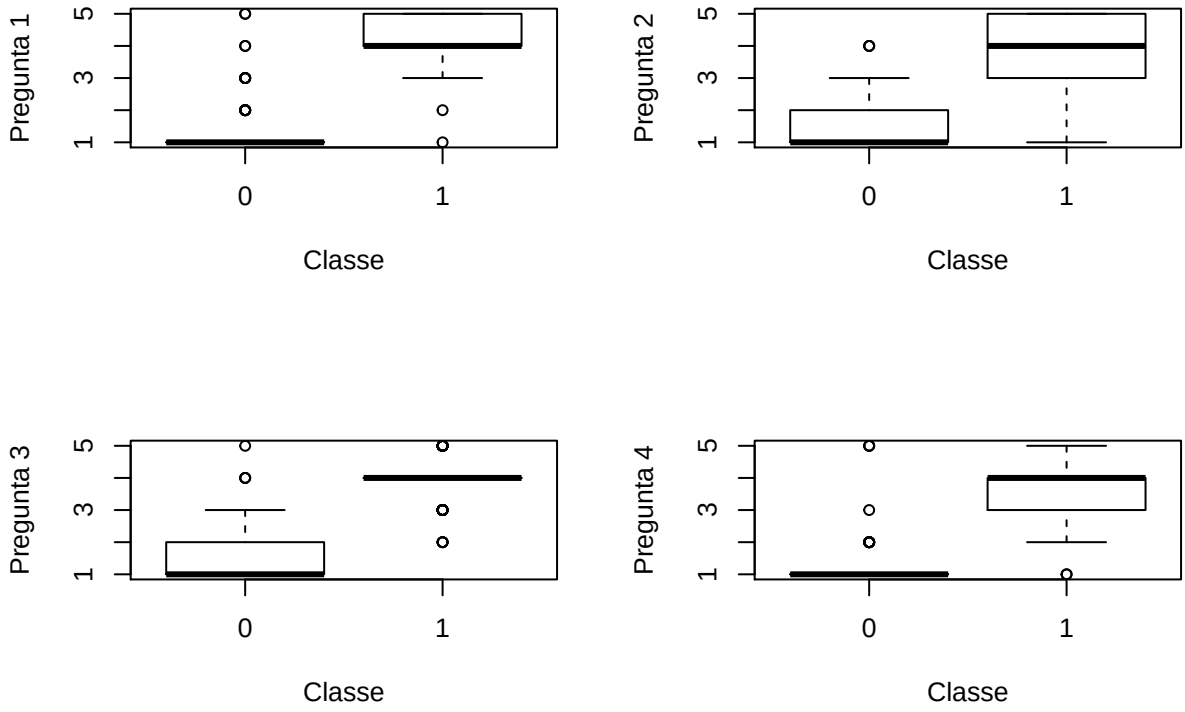
Veiem que les dues classes estan representades igualment en les dades la qual cosa és desitjable.

A més, no hi ha dades codificades com a missing values i, per tant, no ens hem de preocupar del seu tractament. Tampoc s'observen valors estranys amb els quals es puguin haver codificat dades faltants (no hi ha dades fora del rang 1-5 en què s'han de trobar les dades).

Comprovem ara que totes les variables estiguin correlades amb la variable a predir, executant la prova de la chi-quadrat ($\alpha = 0.05$, valor que prendrem durant tot el treball):

```
FALSE
   54
```

S'observa que cap de les preguntes és independent de la classe. Per exemple, si fem un boxplot de les primeres preguntes en funció de la classe a la qual pertanyen les observacions, observem el següent:



Veiem que presenten una separació molt evident entre les respostes d'ambdós grups (depenen de la classe).

Una qüestió molt important és com analitzarem quin dels models és millor. Idealment, partiríem el conjunt de dades en dos grups independents. En el primer, entrenaríem els models i, després, analitzar quin d'ells és millor en el subgrup de validació o fent servir mètodes de resampling. En el segon, calcularíem l'error de test per donar una estimació de l'error de classificació del model escollit.

És important tenir en compte que això fa que perdem dades sobre les que entrenar els models, ja que les dades de test (un terç del total, habitualment) només es fan servir per estimar l'error de classificació. És per això que aquest mètode és molt costós si no tens un conjunt de dades molt extens.

En el nostre cas particular, disposem d'un nombre d'observacions molt reduït (únicament 170 observacions) i considerem que reservar un terç de les dades només per estimar més acuradament l'error de classificació és un luxe que no ens podem permetre. És per això que hem arribat a la conclusió que el més apropiat és fer servir leave-one-out crossvalidation sobre tot el conjunt de dades i prendre l'error donat per aquest resampling method com l'error de predicció de dades futures.

Tractant les dades com a categòriques

Les variables predictores són categòriques, per tant, començarem provant el mètode Naive-Bayes i el Random forest.

Naive - Bayes

Aquest mètode estima les probabilitats posterior de cada classe assumint independència entre les variables predictores i estimant les probabilitats a partir de les dades donades. Fent LOOCV, l'error de test estimat

és de 2.35%.

```
[1] 2.352941
```

D'aquest mètode podem analitzar la hipòtesi d'independència fent servir un altre cop el test de chi-quadrat d'independència per variables categòriques. El farem servir entre tots els possibles parells de variables predictores ($54 \cdot 54 = 2916$):

```
FALSE
 2916
```

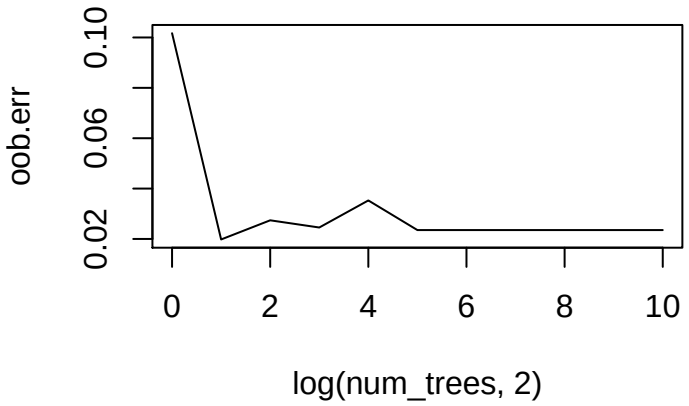
Veiem que la correlació entre les variables és molt alta i, per tant, no són independents (per tots els tests el p-valor és més gran que la significació presa, 0.05). És per això que l'assumpció d'independència del classificador Naive-Bayes no és gaire apropiada.

Random Forest

A continuació, ens ha semblat una bona idea provar d'usar el mètode de random forest. Usant aquest mètode no es fan assumpcions de cap tipus sobre el model probabilístic que segueixen les dades. Així doncs, no fa cap assumpció sobre la independència de les variables predictores, per això potser podem veure si millora el resultat del Naive-Bayes classifier que sí que la fa.

Una altra característica interessant dels random forests és que no cal fer ús de tècniques de resampling per a tenir una estimació de l'error de test. Això es deu al fet que `randomForest` *per se* ja fa ús del bagging (Bootstrapping aggregating). D'aquest forma, obtenim el Out-of-bag error (OOB) que també ens serveix per aproximar l'error de test.

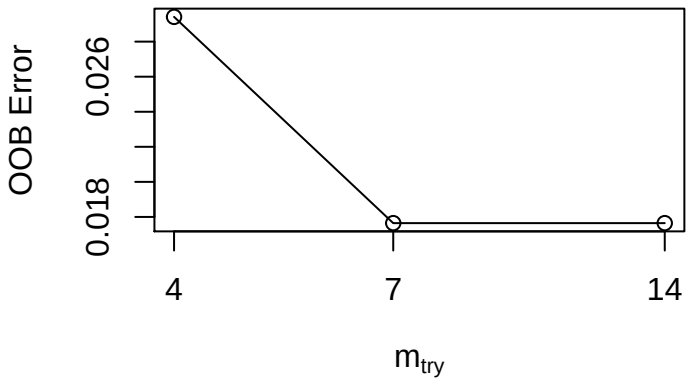
Per a determinar el nombre d'arbres a usar, fem servir un procés iteratiu entrenant diversos random forests amb diferents nombres d'arbres. Farem servir el OOB per determinar el millor nombre d'arbres. Com no és un mètode determinista, fixem la seed a 2048 per a obtenir resultats reproduïbles.




```
      OOB
0.02352941
```

L'error s'estabilitza al 2.35% a partir dels 32 arbres (2^5), mantenint-se constant fins als 1024. Per tant, prendrem un random forest amb 32 arbres, i ara li ajustarem el paràmetre m. Aquesta m és el nombre de variables que es selecciona per a la construcció de cada arbre. Per defecte, als problemes de classificació, s'usa $m = \sqrt{p}$ on p és igual al nombre de variables. La rutina `tuneRF` explora automàticament l'entorn del valor per defecte, tot buscant si l'error OOB millora:

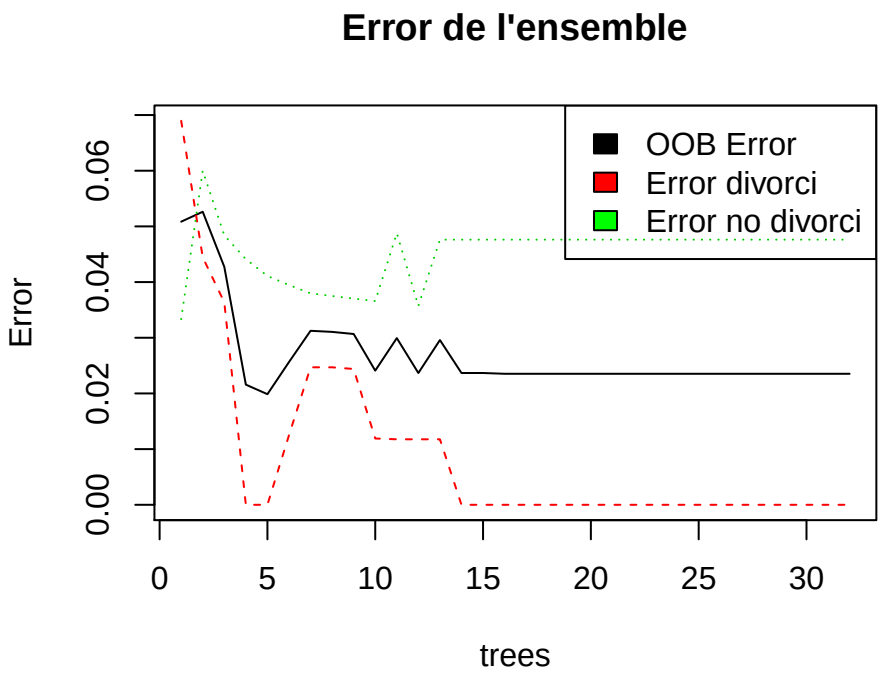
```
mtry = 7  OOB error = 1.76%
Searching left ...
mtry = 4    OOB error = 2.94%
-0.6666667 0.05
Searching right ...
mtry = 14   OOB error = 1.76%
0 0.05
```




```
       mtry   OOBError
4.OOB     4 0.02941176
7.OOB     7 0.01764706
14.OOB   14 0.01764706
```

Com podem veure, el valor per defecte ($m = 7$) sembla ser un òptim local. Així doncs, no el canviarem a l'hora d'entrenar el model definitiu:

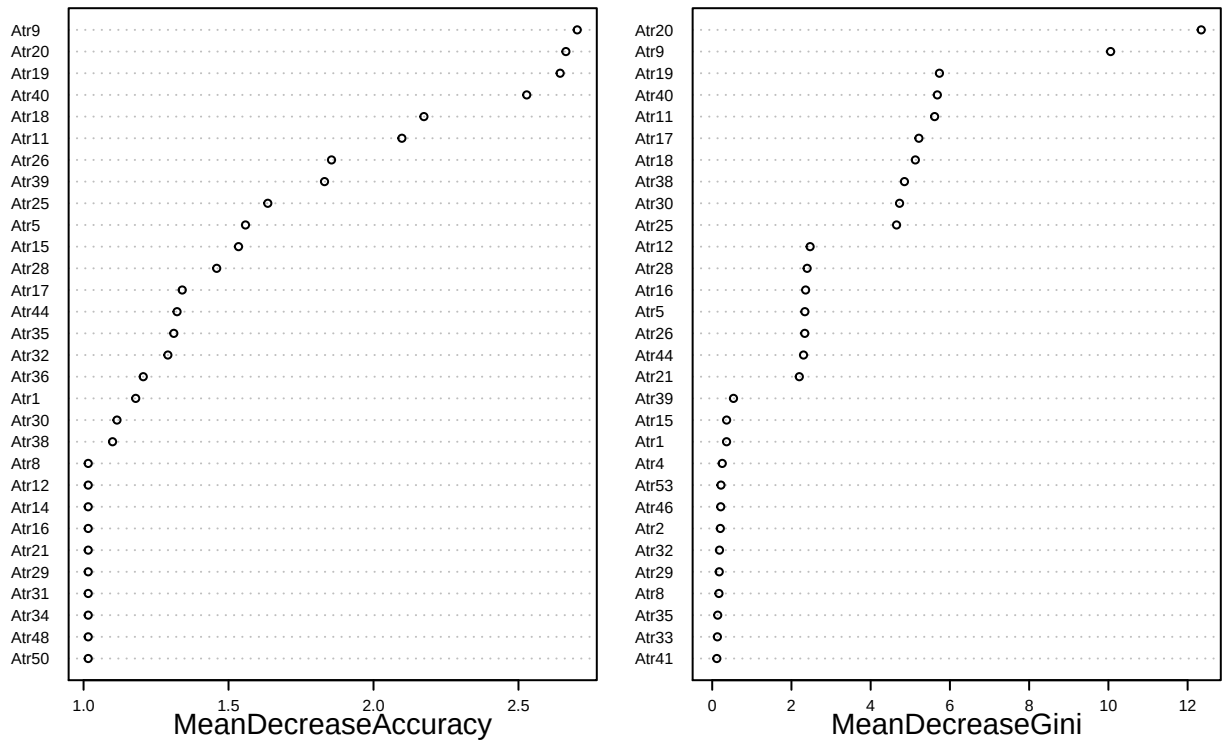
```
Call:
 randomForest(formula = Class ~ ., data = dd, ntree = 32, proximity = T,      importance = T)
               Type of random forest: classification
                     Number of trees: 32
No. of variables tried at each split: 7

        OOB estimate of  error rate: 2.35%
Confusion matrix:
   0  1 class.error
0 86  0  0.00000000
1  4 80  0.04761905
```



Aquest plot ens mostra com varien els errors (OOB i per classes) al anar afegint arbres a l'ensemble. Com es pot observar, fins als 10 arbres la variància és elevada però a partir de llavors l'error va decrementant fins a estabilitzar-se a partir dels 22 arbres.

L'objecte generat per la funció `randomForest` permet utilitzar les rutines `varImpPlot` i `MDSplot`.

La `varImpPlot` ordena les variables predictores en funció de com d'importants són pel que fa al Gini i a l'accuracy del random forest. Aquest càlcul es duu a terme permutant les variables i veient com decrementen les mètriques mencionades respecte a l'ensemble original.

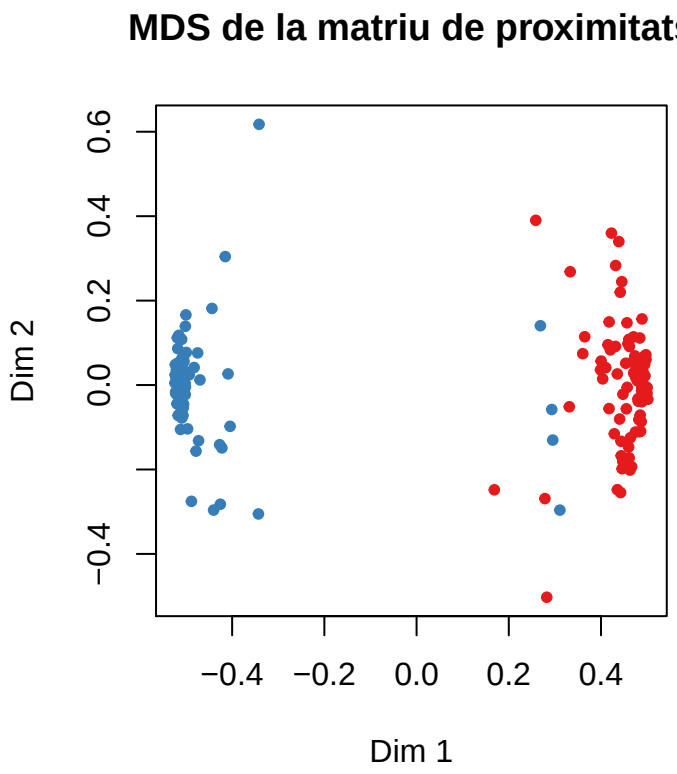


Veiem que tant pel gini com per la accuracy moltes de les preguntes més importants del qüestionari són comunes: 20, 9 i 19, per exemple. Aquests predictors corresponen a les preguntes següents:

- My spouse and I have similar values in trust.
- I enjoy traveling with my wife.
- My spouse and I have similar ideas about how roles should be in marriage.

Té sentit que les preguntes 20 i 19 siguin importants pel bon funcionament d'un casament, ja que tenen a veure amb les idees que tenen els integrants de la parella sobre el matrimoni. També té cert sentit que la pregunta 9 sigui important, ja que viatjant es passa més temps en parella i afloren més fàcilment les emocions.

La rutina `MDSplot` produeix una representació en l'espai euclidià de les observacions a partir de la matriu de proximitats del forest i fent ús la tècnica Multidimensional Scaling (MDS).



Veiem que, en general, hi ha una separació molt satisfactòria de les dues classes, excepte per quatre punts que probablement són els que generen l'error del random forest. Veurem que aquests quatre punts problemàtics es repetiran al llarg del treball.

Fent LOOCV obtenim un error de 2.35% que és igual al OOB error:

```
[1] 2.352941
```

No s'observa millora respecte del classificador Naive-Bayes tot i no fer l'assumpció d'independència. Tot i això, hem pogut analitzar en més profunditat l'efecte dels diferents predictors en la classificació.

Tractant les dades com a numèriques

Com ja hem esmentat anteriorment les nostres dades són categòriques. Tot i això, les categories de cadascuna de les variables categòriques presenten una ordenació natural implícita, ja que estan en una escala de Likert. És a dir les preguntes s'avaluen des de "Molt en desacord" fins a "Molt d'acord" en una escala ordenada. És per aquesta característica que podem tractar aquestes dades com a dades numèriques per aplicar altres models que aplicaríem sobre dades contínues com models lineals generalitzats o el kNN.

GLM

Primer provarem de fer un model lineal generalitzat per classificació binària (regressió logística). L'error de LOOCV és 3.529%:

```
       1
3.529412
```

Aquest mètode dóna un error major que el classificador Naive-Bayes i el Random Forest (2.35%). És probable que es generi over-fitting pel fet d'utilitzar totes les variables. Podríem intentar fer servir la funció `step` per reduir el nombre de variables del model lineal, així i tot, fer això a cada iteració del LOOCV seria molt costós (la rutina `step` tarda molt a executar-se). Per tant, cometrem *pecata minuta* executant `step` sobre un glm entrenat amb totes les dades i n'utilitzarem la fórmula resultant per generar els glm a cada iteració del LOOCV.

```
Call:  glm(formula = Class ~ Atr14 + Atr17 + Atr24 + Atr40, family = "binomial",
    data = dd)

Coefficients:
(Intercept)        Atr14        Atr17        Atr24        Atr40
    -103.90        40.94        42.63       -52.16        39.84

Degrees of Freedom: 169 Total (i.e. Null);  165 Residual
Null Deviance:      235.6
Residual Deviance: 1.567e-08    AIC: 10
```

La fórmula resultant de la rutina és:

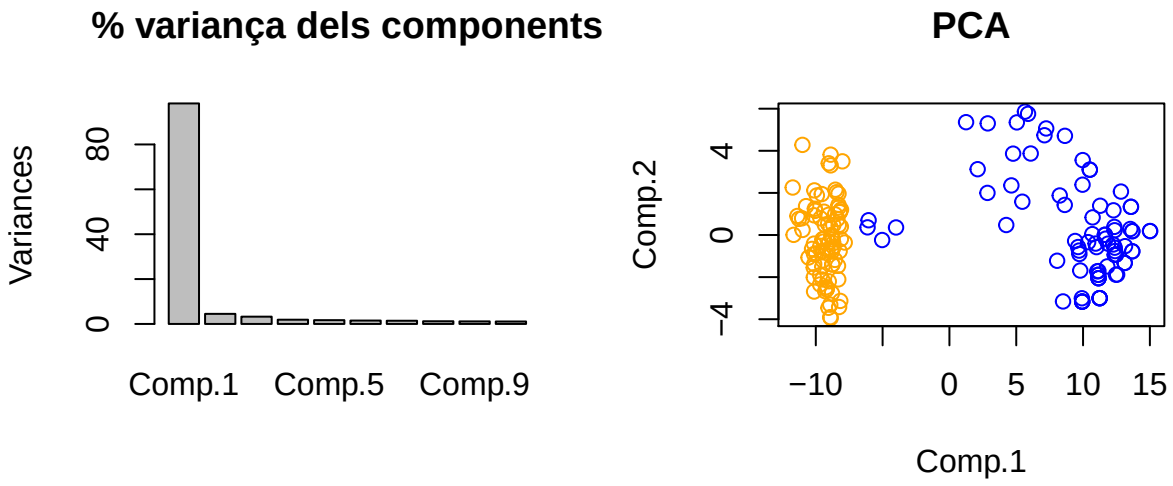
```
glm(formula = Class ~ Atr14 + Atr17 + Atr24 + Atr40, family = "binomial", data = dd)
```

S'ha fet una reducció molt gran del nombre de variables (hem passat de 54 a únicament 4). Probablement, això és degut a un fet que ja hem analitzat prèviament, totes les variables predictores tenen un alt grau de dependència entre elles. L'error obtingut fent servir LOOCV és 1.17%:

```
       1
1.176471
```

Reduir el nombre de variables comporta una millora molt significativa: només dues dades mal classificades.

Una altra forma de reduir el nombre de variables predictores és fent servir PCA i, llavors, fer servir únicament les primeres components com a predictors. Aplicant el PCA sobre totes les dades, obtenim els següents gràfics:



Al primer gràfic, observem que el 99.1% de la variabilitat queda representada amb el primer component principal únicament. Al segon, veiem el plot de les dues primeres components principals i es veu molt bé que hi ha una separació molt clara entre els dos grups, excepte per 4 observacions de parelles no divorciades que queden molt properes a les dades de les parelles divorciades. A continuació, calculem l'error de LOOCV pel glm amb només el primer component principal:

```
1
0
```

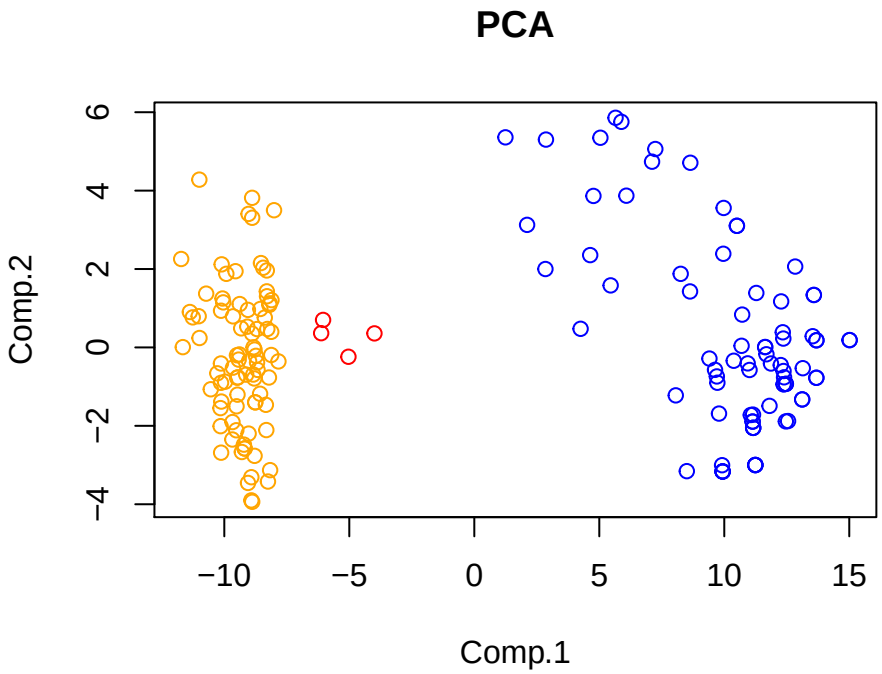
Obtenim un error de test del 0%! No hi ha dades mal classificades. Veiem que el fet de reduir la dimensionalitat i el nombre de variables predictores ha permès disminuir molt l'error de validació.

kNN

Una altra tècnica que podem aplicar per a dades numèriques és el mètode **k-Nearest-Neighbours** (`kNN`) que classifica noves observacions en funció de la classe de les observacions més semblants a aquesta. Aquest model presenta un hiperparàmetre (la k) que pot fer canviar molt les prediccions del model en funció del valor que li donem. Per tant, comparem l'error del LOOCV per cada valor de senars (per evitar empats) de k entre 1 i $\sqrt{n}$:

```
Classification errors in observations:  1 5 6 10
k = 1 LOOCV error: 0.02352941

Classification errors in observations:  1 5 6 10
k = 3 LOOCV error: 0.02352941

Classification errors in observations:  1 5 6 10
k = 5 LOOCV error: 0.02352941

Classification errors in observations:  1 5 6 10
k = 7 LOOCV error: 0.02352941

Classification errors in observations:  1 5 6 10
k = 9 LOOCV error: 0.02352941

Classification errors in observations:  1 5 6 10
k = 11 LOOCV error: 0.02352941

Classification errors in observations:  1 5 6 10
k = 13 LOOCV error: 0.02352941
```

Veiem que per tots els valors de k entre 1 i $\sqrt{n}$ l'error de LOOCV és el mateix (2.35%). De fet, les dades mal classificades per tots els valors de k són les mateixes: 1, 5, 6, 10. Si les representem en el PCA:



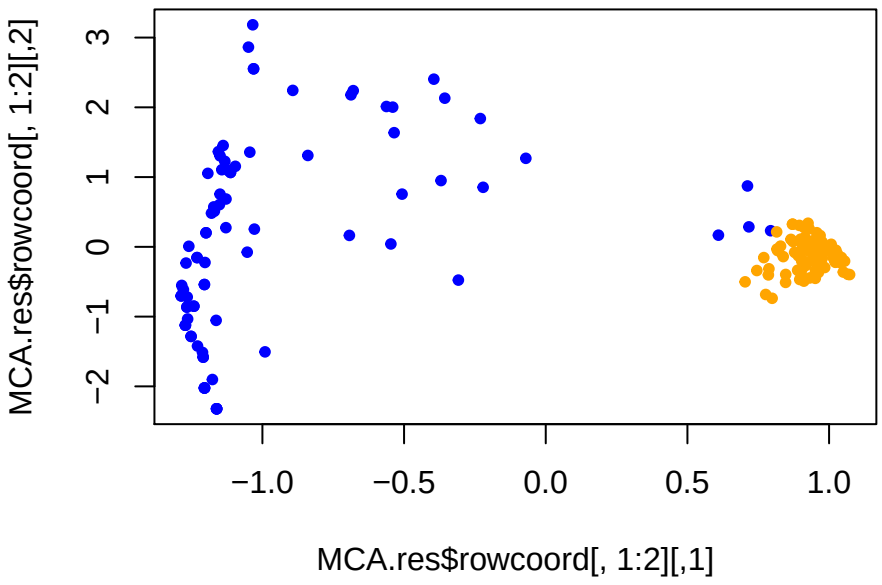
Com era d'esperar l'error que s'ha produït ha estat degut a les 4 observacions que hem comentat anteriorment en el plot del PCA.

MCA

Tot i això, potser traduir a variables numèriques no és del tot apropiat, ja que fent això s'assumeix que les distàncies entre els diferents nivells dels factors són iguals. Per exemple, prenem els tres primers nivells de cadascuna de les variables ("Molt en desacord", "Desacord" i "Indiferent") identificats numèricament com 1, 2 i 3.

En convertir aquestes dades numèricament estem assumint que la "distància" o "diferència" entre respondre "Molt en desacord" i "Desacord" o respondre "Desacord" i "Indiferent" són iguals. És molt possible que l'augment de convenciment necessari per passar de respondre un factor o altre sigui diferent (o fins i tot que canviï per cada pregunta).

Per solucionar aquest problema se'ns acut fer servir l'**anàlisi de correspondències múltiple**, per a obtenir dades en un espai mètric on poder aplicar els mètodes ja utilitzats (kNN i GLM) sense haver d'assumir els problemes d'aquestes conversions numèriques.



Veiem com el resultat del MCA separa clarament les dues categories quasi perfectament. És evident, que 4 observacions de la classe blava (divorciats) són molt a prop de les dades de l'altra categoria i és probable que siguin "problemàtics" i que siguin aquests l'arrel de molts dels errors comesos pels classificadors que hem tingut en compte. Aquests resultats són molt semblants als obtinguts quan hem fet el PCA.

Aquestes noves dades estan en un espai de 216 dimensions. Per reduir dimensionalitat simplement farem servir els eigenvalues de la matriu d'indicadors per veure quantes dimensions són necessàries per representar un percentatge alt (80%) de la inèrcia de la matriu d'indicadors.

```
[1] 33
```

Així doncs, prendrem únicament les 33 primeres dimensions de l'anàlisi de correspondència múltiple.

Models basats en l'MCA

A partir d'aquí, repetirem els dos models que hem generat per dades numèriques (en l'apartat anterior) per tal de veure si apreciem alguna millora aplicant-los al resultat del MCA.

GLM

Primer, farem un glm a partir de totes les variables i hi aplicarem la rutina `step` per tal de reduir el nombre de variables predictores, tal com hem fet abans.

```
Call:  glm(formula = Class ~ V1 + V6, family = "binomial", data = dd.mca)

Coefficients:
```

```
(Intercept)           V1           V6
     181.81      -258.74        19.28

Degrees of Freedom: 169 Total (i.e. Null);  167 Residual
Null Deviance:      235.6
Residual Deviance: 7.423e-08    AIC: 6


        1
0.5882353
```

Per aquest model observem un error de 0.58%. Quan no fèiem servir el MCA, obteníem un error de test del 1.17% així que això suposa una millora.

Abans hem fet servir PCA per millorar aquest resultat, però en aquest cas, això no té gaire sentit, ja que el MCA és l'equivalent del PCA per a dades categòriques. Aplicar MCA al resultat del MCA no té gaire sentit.

kNN

Fent servir les 33 dimensions escollides del MCA:

```
k = 1 LOOCV error: 0.07647059
k = 3 LOOCV error: 0.1
k = 5 LOOCV error: 0.1
k = 7 LOOCV error: 0.1
k = 9 LOOCV error: 0.1117647
k = 11 LOOCV error: 0.1058824
k = 13 LOOCV error: 0.1117647
```

Obtenim uns resultats molt pitjors que abans. Per $k = 1$ obtenim l'error mínim (7.6%). A primera vista, això ens sembla estrany, ja que amb dues i tres dimensions semblava que aniria bé. Si ens fixem en el plot de les dues o tres primeres dimensions del MCA, queda clar que amb un nombre baix de nearest-neighbour els punts quedarien apropiadament classificats (fins i tot uns quants dels "problemàtics").

Escollint només les tres primeres dimensions (les del plot)

```
k = 1 LOOCV error: 0.005882353
k = 3 LOOCV error: 0.005882353
k = 5 LOOCV error: 0.005882353
k = 7 LOOCV error: 0.01764706
k = 9 LOOCV error: 0.02352941
k = 11 LOOCV error: 0.02352941
k = 13 LOOCV error: 0.02352941
```

Veiem que la nostra hipòtesi és correcta, amb menys dimensions el kNN funciona molt millor. Per $k = 1, 3, 5$ l'error és de tan sols 0.58%. És probable que les dimensions que representen un percentatge menor de la variabilitat representin el soroll de les dades, mentre que les primeres dimensions són les reals. Com és d'esperar, el soroll afectaria negativament a la predicció de les classes.

Conclusions

En aquest treball hem tractat de classificar les parelles en un conjunt de dades entre les divorciades i les casades. Per fer-ho hem fet servir les dades d'un qüestionari que hem tractat des de diferents punts de vista. Primer les hem tractat com a dades categòriques i hem aplicat el classificador de Naive-Bayes i un random forest. A continuació, les hem tractat com a numèriques, ja que hi ha una ordenació natural de les dades i hem aplicat una regressió logística i un k-nearest Neighbours. Per la regressió logística hem reduït el nombre de variables predictores fent servir la rutina `step` i l'anàlisi de components principals per tal de reduir l'over-fitting i disminuir així l'error obtingut en fer LOOCV. També hem decidit de fer servir Multiple Correspondance Analysis per tal de convertir les dades a numèriques d'una forma més consistent (les diferències entre els diferents nivells poden ser diferents). Sobre aquest nou espai mètric hem tornat a aplicar els mètodes emprats prèviament, la regressió logística i el kNN. Els errors de LOOCV obtinguts per cadascun d'aquests mètodes es resumeixen a la següent taula:

Dades	Mètodes	Error
Categòriques	Naive Bayes	2.35%
	Random Forest	2.35%
Numèriques	GLM	3.53%
	GLM + Step	1.17%
	GLM + PCA	0%
	kNN	2.35%
MCA	GLM + Step	0.58%
	kNN (33 dims)	7.65%
	kNN (3 dims)	0.58%

Amb aquests resultats sembla que el model correcte seria una regressió logística a partir del primer component principal del PCA. Tot i això, hem de ser conscients que teníem molt poques dades per fer l'estudi dels nostres models i les estimacions del percentatge d'error estan fets a partir d'una mostra molt petita. A més, les dades que hem obtingut són d'enquestats de Turquia per la qual cosa s'hauria d'investigar com afecta això a les prediccions per a dades de països diferents amb cultures diferents (percentatge de la població religiosa, confessions...).

Ens ha sorprès la precisió de les prediccions, ja que, a priori, semblaria complicat determinar l'estat d'una parella a partir d'un simple qüestionari. Ha estat un treball molt interessant i ens ha permès utilitzar les eines apreses a l'aula d'una forma molt pràctica i realista.

Referències

Yöntem, M , Adem, K , İlhan, T , Kılıçarslan, S. (2019) - DIVORCE PREDICTION USING CORRELATION BASED FEATURE SELECTION AND ARTIFICIAL NEURAL NETWORKS. Nevşehir Hacı Bektaş Veli University SBE Dergisi, 9 (1), 259-273. Retrieved from [Web Link]

Johnson, R.A, Wichern, D.W - Prentice-Hall (2014) - Applied multivariate statistical analysis - ISBN: 9781292037578 Retrieved from [Web Link]

Peña, D - McGraw-Hill/Interamericana de España, S.L (2013) - Análisis de datos multivariantes - ISBN: 9788448191849 Retrieved from [Web Link]

Breiman, L, Cutle, A (2004) - Random Forests Retrieved from [Web Link]