Memòria pràctica 8

Josep Marc Mingot//, Gabriel Reines

6 de gener de 2013

Resum

Índex

1 Introducció

[Explicació del problema i lllista de variables a usar.] La KDD Cup és una competició anual de Mineria de Dades organitzada per l'ACM (Special Interest Group on Knowledge Discovery and Data Mining). Cada any milers d'equips de tot el món competeixen per donar solució a un problema de mineria de dades patrocinat per una empresa o organització. En aquest treball ens hem plantejat donar una solució al problema plantejat per la KDD Cup del 2006, patrocinada per la Universitat de Nuevo Méjico.

El problema que és va plantejar en aquesta edició consistia en la detecció de embolies pulmonars (PE) a través de dades extretes de un CAD 3-d del pacient. Les PE ocurreixen quan una artèria del pulmó queda obstruida. Malgrat que aquesta obstrucció molts cops no és letal, és la tercera causa més comuna de morta als EUA amb almenys 65000 casos cada any. El repte està en poder diagnosticar al pacient el més aviat possible. Això no obstant no és gens senzill ja que els símptomes dels PE són molt pareguts a altres patologies molt comunes. Per detectar PE una tècnica molt recent és l'anomenada CTA (Computed Tomography Angiography) que consisteix en pendre moltes imatges dels pulmons cada una de elles representant una secció del pulmó. Aquestes imatges són processades computacionalment per extreure'n de cada una una variables i *candidats* a PE. Entenem per candidat a aquell punt del pulmó que pot ser un PE (encara que també podria ser altres artifactes del pulmó). El repte consisteix en classificar els PE donades un conjunt de variables ja extretes de les imatges.

La base de dades està formada per 38 pacients positius i 8 negatius. En total tots aquests pacients donen lloc a 3038 candidats a PE que són les files de la nostra base dedades. Cada candidat conté 116 variables extretes de les imatges de CAD dels pulmons. Les tres primeres variables són les coordenades x, y i z del candidat. La resta de variables poden ser dividides en 3 grups: aquelles que corresponen a la distribució d'intensitat dels voxels que formen el candidat, les distribucions d'intensitat dels voxels del voltant del candidat i les que descriuen la forma 3-D.

Finalment fem un breu incís sobre el marc de treball que hem configurat per avaluar els classificadors. Hem dividit les dades en un conjunt d'entrenament (70%) i un de test (30%). El conjunt de test és l'usat per avaluar l'error en els classificadors. En alguns casos s'ha usat la tècnica de la validació creuada *k-fold* per donar més robustesa a les estimacions de l'error. En ells, s'ha dividit el conjunt de dades en 5 grups: 4 s'han usat per entrenar i 1 per avaluar donant un total de 5 mesures de l'error. Finalment per avaluar l'error dels classificadors usem 2 mpetriques diferents. La probabilitat d'error directament sobre el conjunt de test i l'àrea sota la curva ROC (AUC). La diferència princpial entre aquestes dos mètriques és que mentre la primera ens diu la probabilitat de classificar un candidat aleatori com a correcte, la segona ens estima la probabilitat de classificar un candidat positiu aleatori com a correcte. En aquest sentit l'AUC ens dona una mesura més robusta que la probabilitat d'error en conjunts no balancejats.

2 Visualització de dades

El primer pas en qualsevol problema de classificació és entendre les dades que se'ns han donat: realació entre les diferents variables, rànking de quines són potencialment més importants i distribucions d'aquestes. Per aquest motiu en aquesta secció pretenem donar a través de l'anàlisi visual, una serie de propietats sobre les variables que influeixen en el problema.

Comencem estudiant la correlació entre elles. Al tenir 116 variables l'eina més adequada per visualitzar l'autocorrelació entre elles és plasmar gràficament la matriu de correlació. En la figura 1 observem la matriu on cada casella representa un valor de la matriu i com més fosc més proper a 1. Hi ha variables que estan completament correlacionades (per exemple `neighborhood intensity feature 2` amb `neighborhood intensity feature 8`). Per eliminar-les, fixem un nivell a partir del qual considerem que les variables estan correlacionades i n'eliminem aquella que té una mitja de correlació més alta amb les altres. Després d'aquest procés i fixant un tall a 0.85 passem de 116 variables a 91 variables. Podem observar la matriu d'autocorrelació neta en la figura 2.

Després de netejar les variables autocorrelacionades, fem un anàlisi exploratòria d'algunes variables. De les variables ens interessa coneixer la seva distribució condicionada a la classe, boxplots segons classe, scatter plots 2 a 2 amb altres variables (per veure possibles parelles de variables que separin). Podem obtenir tots aquests descriptius de les variables en una sola imatge. En la figura **??** observem aquests descriptius per les variables de posició (coordenades x,y i z del PE) i la mida del PE candidat (amb el nom de `V5`). Observem com l'*scatter* de les coordenades de posició ens dibuixen els plans de tall d'uns pulmons com calia esperar. Tanmateix observem també que no és descriminador la posició del candidat per dir si és o no PE. Respecte a la variable "mida"si que veiem que és més discriminador: els scatters amb les coordenades ens consegueixen separar prou bé els 1 dels 0.

Finalment en la figura **??** podem observar els mateixos descriptius comentats anteriorment per a les 7 variables més significatives del model segons el *CAT score* (ja en donarem més detalls en l'apartat LDA). Observem per anàlisis visual que cap de les distribucions condicionades (primera columna de la imatge) s'acosta a una distribució gaussiana, però les distribucions condicionades a les classes són diferents (no s'acaba de poder apreciar ja que motrem la freqüència enlloc de la densitat). Per altra banda, els boxplots condicionats (primera fila) en mostra com efectivament ens donen distribucions diferents segons la classe. Com més diferents siguin les distribucions condicionades a la classe més senzill en serà poder separar (no en va estem mostrant aquí les nostres top 7 variables).

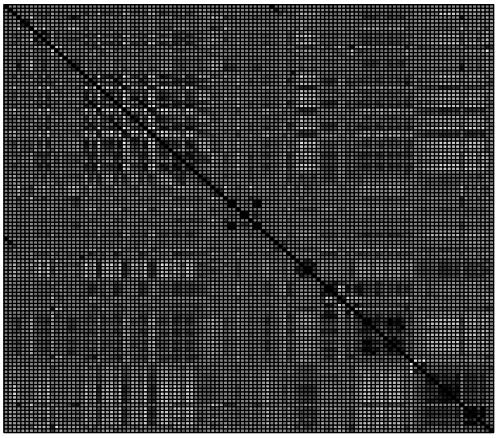

Figura 1: Autocorrelació de les 117 variables.

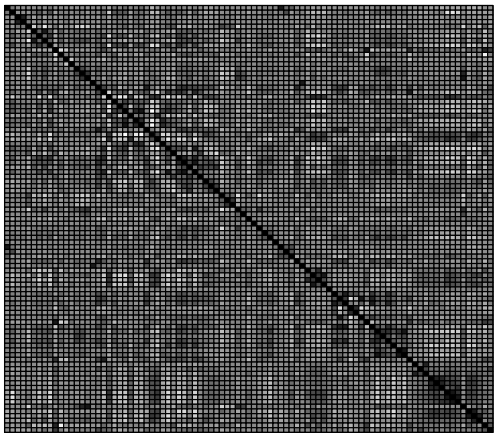

Figura 2: Autocorrelació de les 91 variables.

3 Reducció de dimensionalitat

4 Classificadors

4.1 DDA, LDA, QDA

En aquest aprtat avaluarem els resultats dels classificadors bastats en discriminants. Concretament hem estudiar el DDA (*Diagonal Discriminant Analysis* o *Naive Bayes* on la matriu de covariàncies diagonals o, equivalentment, que les variables del model són independents), el LDA i QDA. Analitzem també diversos escenaris en funció de les variables d'entrada dels models . De tots ells en donem la probabilitat d'error en el conjunt de test i conjunt d'entranament i quan és possible la curva ROC en el conjunt de test.

Comencem mostrant una comparativa entre els tres classificadors aplicats sobre les 91 variables (n'hem extret les no correlacionadaes tal com hem comentat en la secció de visualització de dades). En la taula 1 podem observar els errors de classificació dels tres algorismes en els conjunts de train i test. DDA, que fa la hipotèsi més forta de independència entre les variables obté el pitjor resultat en els dos conjunts. Tant LDA com QDA obtenen errors amb ordres de magnitud similars. Tanmateix QDA sembla produir una mica més de *overfitting* al tenir un error menor en el conjunt d'entrenament i un superior al de test respecte LDA. En sentit global podem dir que LDA obté millors resultats per aquest problema de QDA.

	error_train	error_test
DDA	0.18	0.22
LDA	0.09	0.10
QDA	0.07	0.11

Taula 1: Comparativa dels classificadors sobre les 91 variables no classificades

En la figura 3 hem inclòs també la curva ROC per als classificadors DDA i LDA per poder comparar-los millor.

4.1.1 Selecció i projecció de variables

En la segona part de l'ànalisis d'aquests classificadors, hem avaluat el LDA (per ser el que millors resultats ha obtingut) per diferents conjunts de variables. Pretenm aquí veure l'efecte dels classificadors en dos nous subconjunts de variables. Per una banda, usant les variables obtingudes per PCA i ICA (projecció). Per altra, usant només aquelles variables més significatives (selecció), on ja detallarem qué entenem per més significatives.

PCA i ICA

La segona metodologia empleada per la selecció de varibles és la de selecció de les més significatives. La selecció de variables més significatives sol ser una fase

molt important en l'entrenament de classificadors per dos motius: per una banda redueix l'error de generalització i per altra disminueix el cost computacional de l'entrenament. Existeix una àmplia bibliogràfia sobre diferents metodologies de selecció de variables, tant seleccions genèriques (tipo filtres, tal com hem fet al eliminar les variables correlacionades) com seleccions espcífiques per cada classificador (*wrapper*). En el nostre cas hem empleat el marc teòric del classificador LDA per puntuar cada una de les variables mitjançant els *CAT scores* (una explicació detallada del mètode és pot trobar a [1]). En la imatge 4 podem observar les 10 variables més significatives així com la magnitud de la seva importància i el signe de les seva contribució (si el seu augment contribueix a una o una altra classe). Per tal de saber quin era el nombre de varibales més adient per usar, em fet una gràfica del error comés en el test set segons els nombre usat. Aquesta gràfica (figra 5) ens mostra que per 43 variables obtenim l'error en el test menor. A partir de llavors l'error incrementa de nou.

En la taula 2 es pot veure els resultat de tots els subconjunts de variables usats: `LDA` per al classificador amb les 91 variables, `LDAPCA` i `LDAICA` per als classificadors amb variables PCA i ICA respectivament i `LDAselect` per al classificador usant només les 43 variables més importants. [ANALISIS ICA I PCA]

	error_test
LDA	0.10
LDA_PCA	0.00
LDA_ICA	0.00
LDA_selec	0.10

Taula 2: Comparativa de LDA aplicat sobre diversos conjunts de variables

4.2 k-NN

4.3 SVM

4.4 Xarxes Neuronals

Presentem aquí els resutlats de l'entrenament de l'entrenament de les Xarxes Neuronals (NN). En primer lloc, hem tingut en compte la dependència de les NN respecte el seus valors inicials per lo que hem canviat el marc de treball per fer-lo més robust. Hem també usat diferents mètodes per determinar els paràmetres òptims de les xarxes i presentem també els resultats de classificació i curves ROC per diferents conjunts de variables usats.

Per entrenar les xarxes neuronals hem canviat una mica el nostre marc de treball format, fins ara, per un conjunt d'entrenament i un de test. Degut a que les xarxes neuronals tenen una dependència elevada amb els seus valors inicials, hem optat per usar la tècnica de la validació creuada *k-fold*. Aquesta tècnica consisteix en partir el conjunt total de dades en k parts. Aleshores, s'entrena el classificador en $k-1$ i s'evalua en la restant. D'aquesta manera és pot usar el mateix conjunt de dades per obtenir fins a k mesures de l'error sobre un conjunt de test. Respecte l'entrenament de les xarxes, hem seguit la filosofia

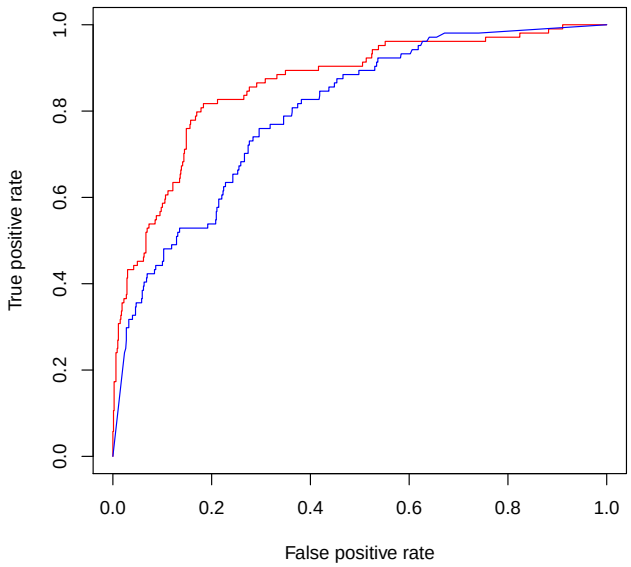

Figura 3: Curva ROC de LDA (en vermell) i DDA (en blau).

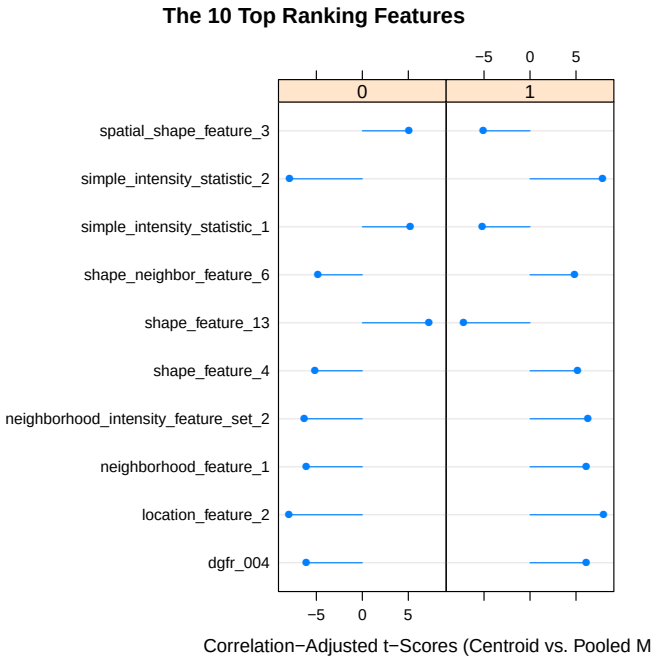

Figura 4: Ranking de les 10 variables més significatives i la seva contribució a cada una de les classes.

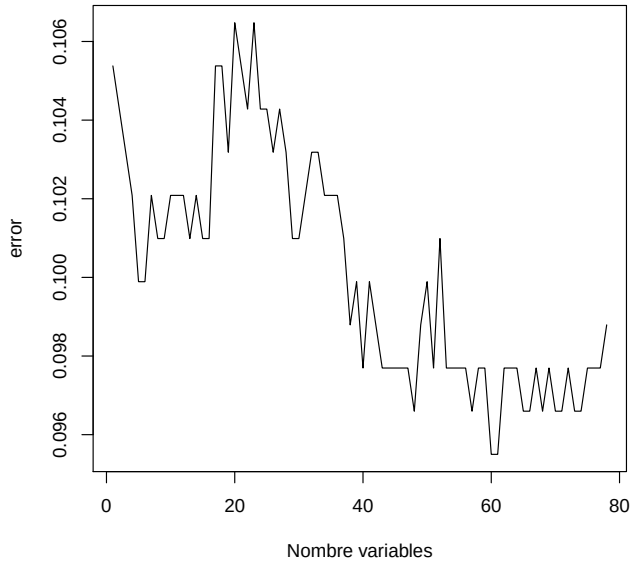

Figura 5: Error en el conjunt de test en funció del nombre de variables usades (oredenades segons importància de CAT score).

general suggerida a [2]. Allí se'ns proposa entrenar les xarxes neuronals amb només una capa oculta. Per seleccionar el nombre de unitats d'aquesta capa, s'introdueix un terme de regularització (*shrinkage*) que ens penalitza tenir molts paràmetres. Així doncs, la metodologia usada ha estat entrenar les xarxes amb 8 unitats imaginàries i determinar per validació creuada el valor òptim del terme de penalització. Aquest parametre regularitzador ja se n'encarregarà de posar a 0 les neurones de la cap intermitja que facin falta.

En la figura 6 podem observar com varia l'error en el conjunt de test (mitja dels k entrenaments) en funció del terme de regularització. En totes elles estem usant 8. Després de diverses execucions probant diferents valors, obtenim que l'ordre de magnitud del valor òptim és 10^{-2}. Per aquests paràmetres descrits podem observar en la figura 8 la curva ROC per les diferents realitzacions de *5-fold* validació creuada. Obtenim una mitja de 0.80 per l'AUC. Pel que fa a l'error comés l'òptim descrit obtenim 0.141 que és lleugerament més elevat que en LDA.

Finalment i paral·lelament al que hem estat fent amb els altres classificadors anem a estudiar com es comporten les xarxes neuronals en funció de les variables usades. En aquest cas només mirarem l'evolució de l'error en les NN en funció del nombre de components principals usades (provinents de la projecció PCA) i en compararem el resultat amb usar totes les variables. En la figura **??** podem veure l'error en el conjunt de test (després de passar per validació creuada 5-*fold*) en funció del nombre de components principals usades. Podem veure que amb 7 components principals ja obtenim un error de 0.16, molt pròxim al

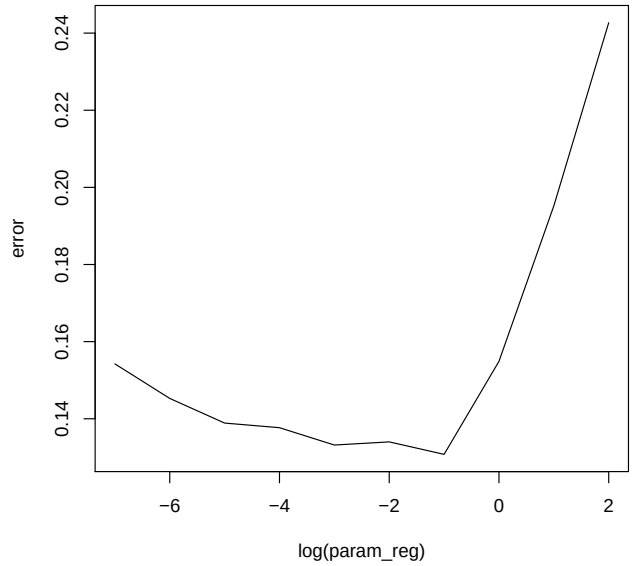

Figura 6: Error en el conjunt de test en funció del valor del paràmetre de regularització emprat en una nn d'una capa oculta i 8 unitats amagades. La variació del parametre regularitzador s'ha realitzat en escala logarítmica.

obtingut amb les xarxes neuronals amb totes les variables (91!).

4.5 Arbres

4.6 Random Forest (opcional)

Els *Random Forest* (RF) són uns classificadors que han guanyat molta reputació recentment degut a la seva alta precisió. En aquest apartat comencem describint-ne breument el seu funcionament. Tal com hem fet amb els altres classificadors, busquem els seus paràmetres òptims minimitzant l'error en un conjunt de test.

Els RF fan ús del arbres de decisió combinats a través del *bagging*. L'algorisme d'aprenentatge és el següent (considerant un conjunt de N exemples i D variables):

1. Escollim un subconjunt d'entrenament (selecció aleatòria amb remplaçament) n inferior al total d'exemples N.

2. Seleccionem aleatòriament un subconjunt de variables d inferior al total D.

3. Amb aquests elements entrenem un arbre de decisió deixant-lo créixer fins al final (sense poda).

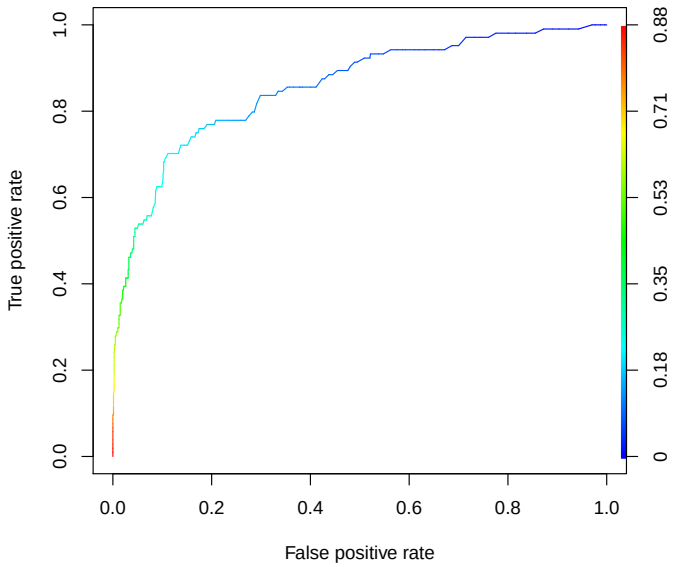

Figura 7: Curves ROC per la NN amb 8 neurones en capa oculat, amb un terme de regularització de 0.05. S'obté una AUC mitja de 0.85.

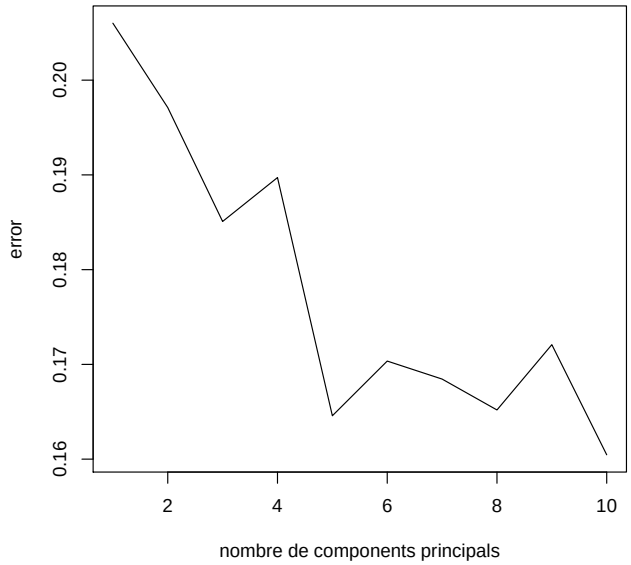

Figura 8: Evolució de l'error en el conjunt de test en funció del nombre de components principals usades.

4. Repetim el procés per un nombre *mtree* d'arbres.

5. Finalment, la decisió de classificació de una nova instància es pren per vot majoritari sobre tots els arbres.

Aquest algorisme s'ajuda de l'aleatorització i combinació de classificadors (tan pel conjunt d'entrenament com per les variables usades en cadascun) per tal de millorar el compromís mitja-variància de classificació.

Les dos proves que hem dut a terme per acabar de refinar els RF són precisament per a concretar el nombre de variables a usar, d i el nombre d'arbres a entrenar, *mtree*. En la figura 9 presentem les gràfiques de l'error en funció d'aquests dos parametres. En cada gràfica s'ha deixat estàtic el paràmetre que no s'estava estudiant [1]. En base a aquestes gràfiques hem escollit usar 17 variables i 50 arbres.

Finalment presentem pels valors òptims obtinguts les curves ROC després de la validació creuada *k-fold* en la figura 10. Els RF també ens proporcionen una mètrica de la importància de les variables que podem contrastar amb la obtinguda a través dels *CAT scores* dels DA. Aquesta mètrica està basada en la impuritat de Gini. Segons aquesta, les 10 variables més importants són:

	variable	Guany Gini
1	simple_intensity_statistic_2	27.50
2	size_normalized	20.70
3	spatial_shape_feature_3	14.93
4	dgfr_006	13.64
5	location_feature_2	12.48
6	neighborhood_feature_1	11.61
7	simple_intensity_statistic_1	11.15
8	dgfr_004	10.94
9	location_feature_1	10.27
10	neighborhood_intensity_feature_set_2	8.92

Taula 3: 10 variables més importants segons el criteri d'impuritat de Gini extrets a partir de l'entrenament dels extitrandom forest

5 Conclusions

A Primer Apèndix

Referències

[1] Miika Ahdesmaki and Korbinian Strimmer, *Feature selection in omics prediction problems using CAT scores and False Nondiscovery Rate Control*, The Annals of Applied Statistics, Vol. 4, 2010.

[1] Un *tunning* dels paràmetres més precís hagués estat anar iterant els resultats obtinguts simultàniament per obtenir l'òptim dels dos paràmetres a la vegada.

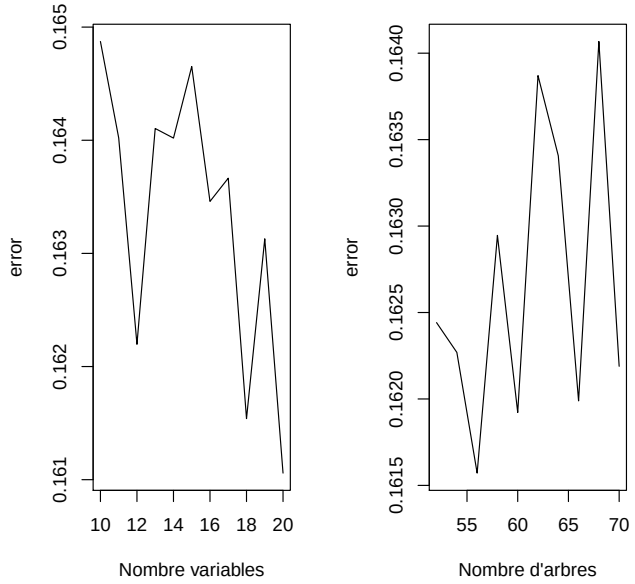

Figura 9: Evolució de l'error en el conjunt de test en funció del nombre de components principals usades.

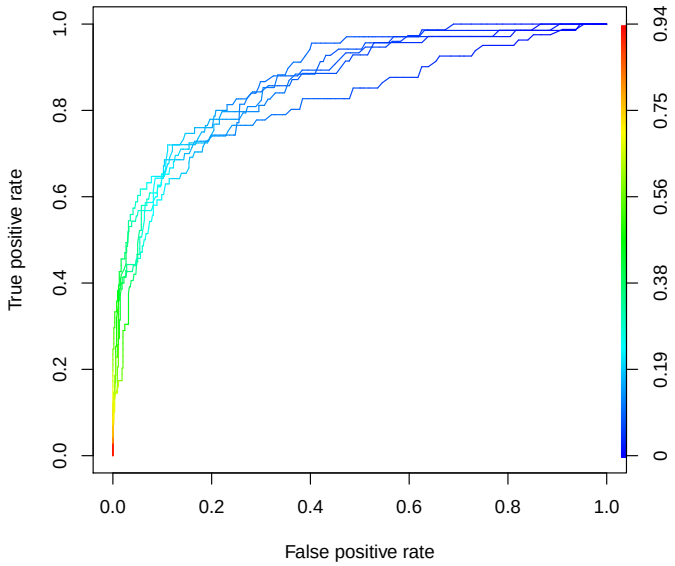

Figura 10: Curves ROC pels 5 conjunts de validació creuada per als paràmetres òptims obtinguts (17 variables per arbre i 50 arbres). Mitja AUC de 0.86.

[2] Trevor Hastie, Robert Tibshirani, Jerome Friedman, *The elements of statistical learning*, Springer, 2nd ed., 2009.