Bank telemarketing: a data-driven analysis

Cristina Aguilera, Jesús M. Antoñanzas
Aprenentatge automàtic 1, Ciència i enginyeria de dades
UPC

Juny 2019

Contents

1 Introducció

En aquest projecte s'exposa el treball d'anàlisi i predicció realitzat a partir de les dades contingudes en el *dataset* anomenat "Bank Marketing". Les dades que trobem en aquest dataset van ser recollides directament a partir d'una campanya de màrqueting realizada per una institució bancària portuguesa. L'objectiu de la campanya era que els seus clients contractessin un dipòsit bancari a llarg termini.

La campanya es va realitzar des de maig del 2008 fins el novembre de 2013 i es troben totes les dades ordenades temporalment.

Disposàvem de dos conjunts de dades diferents. En el més antic, només es feien servir 17 variables relacionades amb les dades personals del client i dades sobre la pròpia campanya de màrqueting. En el darrer, s'hi van afegir 5 variables més de caràcter socioeconòmic. En estudis previs, van arribar a la conclusió que aquestes noves variables augmentaven significativament la predicció. Per aquest motiu, s'ha decidit centrar l'anàlisi en el segon conjunt de dades.

El nostre objectius per aquest projecte és, principalment, aconseguir un model de predicció amb mínim error que es centri principalment en minimitzar l'error de les prediccions amb resposta "yes". Com a objectiu secondari, volem veure quines són les variables que més afecten a l'hora de la predicció, i si aquestes són les darreres variables afegides al dataset (socioeconòmiques).

2 Treballs previs

El set de dades ha estat analitzat previament dos cops. L'anàlisi del data set més antic va ser publicat el 2011 (Sergio Moro, Paulo Cortez, Raul M. S. Laureano) i el de les més recents, el 2014 (Sergio Moro, Paulo Cortez, Paulo Rita). Abans de pre-processar, es van agafar 150 variables diferents, escollides per experts de la banca. Després, però, es va efectuar un procés de selecció de les variables més rellevants, i en van quedar 22. Amb aquestes, les dades es van modelar amb 4 tècniques: regressió logística, arbres de decisió, SVM i NN, amb el millor resultat donat per la xarxa neuronal, capaç d'aconseguir un 79% dels subscriptors originals "contactant" la meitat dels clients.

Les mètriques utilitzades per mesurar la capacitat dels models (*Area Under the ROC Curve i ALIFT*) són més avançades i específiques del que farem servir nosaltres. A més, hi han variables, com per exemple la direcció de la trucada (inbound o outbound) o experiència de l'agent de marketing, que a partir de dos mètodes d'extracció de coneixement van resultar ser molt importants per a la classificació amb millors resultats. Aquestes variables no estan disponibles al nostre set de dades, per la qual cosa i com era previsible esperem resultats que no siguin comparables.

Malgrat això, els estudis són una bona referència per començar i una font d'inspiració.

3 Treball exploratori

L'exploració del set de dades comença amb l'anàlisi de les variables. Per a cadascuna, mirar valors mancants i buscar patrons, entendre el paper que juga al set de dades, pensar si cal canviar el tipus (factor, ordinal o numèrica) i explorar les relacions que té amb les altres és un treball que cal realitzar si es volen metodologies consistents i amb sentit i, potser, bons resultats.

L'estudi de les relacions entre variables consisteix en visualitzacions per parelles de variables que semblen correlacionades.

Aquesta exploració seguidament ha de ser complementada, si cal, amb eines d'imputació, de balanceig de dades o d'altres per tal d'enllestir el conjunt de dades i tenir-lo preparat per a la modelització.

Cal mencionar que les decisions preses durant el pre-procés no són l'única opció. És a dir, hi ha camins alternatius que es poden prendre enlloc dels que nosaltres hem considerat adients, ja que poden ser iguals o més vàlids. Per aquesta raó, hem documentat les diverses opcions disponibles en cada moment.

3.1 Pre-procés

Després d'analitzar detalladament cadascuna de les variables i les seves característiques, es comentaran els aspectes més destacadas d'aquestes. Cal destacar que no hem trobat valors anòmals a cap variable.

- **Age**. S'ha trobat que la variable age agafa un rang d'edat molt gran, amb persones des de 17 anys fins 98. S'ha cregut convenient categoritzar aquesta variable per rangs d'edat. Així aconseguirem representar les diferents generacions que suposem tindran característiques i comportaments semblants. Hem basat aquesta categorització en l'article "Average net worth by age" en el qual estudia el valor net mitjà dels americans i els separa en diferents rangs d'edats en el que aquest número és aproximadament el mateix. Realment aquest anàlisi és extrapolable a la població portuguesa, ja que encara que tinguin uns sous més baixos que els americans els rangs seguiran sent els mateixos. Malgrat tot, aquesta classificació té el desavantatge de no estar balancejat. Per això, també podem plantejar la idea de dividir les edats per quartils. Aconseguiriem tenir el mateix nombre de persones en cada rang, però aquestes no compartiran característiques i segurament no es comportin igual.
- **Job**. Es troba que els enquestats es distribueixen en 12 sectors professionals diferents entre els que s'inclouen administratius, serveis, jubilats, estudiants i emprenedors. Inclou 330 persones de les quals es desconeix el seu treball. Més endavant, podría ser una bona idea imputar-les, sobretot a partir del nivell d'educació, ja que hem observat una clara associació entre les posicions de treball més baixes amb gent que té uns nivells d'estudis baixos. Cal tenir en compte, però, que en proporció tenim moltes més observacions de posicions de treballs inferiors com admin i blue-collar.
- **Marital**. Com que aquesta variable té poc valors nuls (80 persones), es podrien intentar imputar a partir d'altre variables que creiem tenen relació com és age o housing.
- **Education**. Conté 1731 valors desconeguts que podríem imputar a partir d'altre variable, però són forces observacions i potser afegim algun tipus de biaix en aquest procés.
- **Default**. Aquesta variable diu si la persona ha incomplert el termini de pagament en algun dels seus crèdits. Conté 8597 valors desconeguts, sent la variable amb més nombre de valors mancants. A més està molt desproporcionada, ja que tenim només 3 persones que han respòs yes. Per tant, hem decidit eliminar aquesta variable ja que no seria útil en el procés de modelització.
- **Housing + Loan**. Les dues variables indiquen si el client té algun crèdit de tipus personal: lloguer i crèdit personal, respectivament. Remarcar que ambdues tenen 990 valors mancants i coincideix que són les mateixes persones. Caldria tractar ben acuradament aquest parell de variables, ja podrien influir bastant per aquesta característica.
- **Contact**. Hi havia dues maneres de contactar per telèfon o mòbil. No hi ha res a destacar.
- **Month + Day of week**. Són dues variables que ens indiquen la data en la que el client va ser contactat. La primera té 10 factors, ja que no inclou ni gener ni febrer, i la segona 5 categories que estan prou equilibrades.
- **Duration**. A priori, la duració de la trucada no la sabem, i per tant, aquesta variable cal ser eliminada del conjunt: és fàcil veure que si una trucada dura molt el més probable és que la persona vulgui contractar un depòsit. La guardem, però, per poder realitzar benchmarking.
- **Campaign**. El número de cops que un individu ha sigut contactat en aquesta campanya. En visualitzar hem observat que cap persona que ha sigut contactada més d'aproximadament 25 cops ha contractat el dipòsit i que la majoría d'individus que ho han fet han sigut contactat poques vegades. Clarament, aquesta és una variable numèrica.

- **Pdays**. Aquesta variable està molt desbalancejada: la majoria d'individus no han sigut contactats anteriorment (marcats com '999') i els que sí ho han sigut ho van ser fa pocs dies (com a molt 27 i en mitjana 6). Aquest desbalanceig suggereix una transformació de la variable a categòrica amb tres nivells: "mai contactat", "contactat fa més de 7 dies" i "contactat fa menys de 7 dies" . Això, però, no ho hem fet, ja que també té sentit que les persones que mai han sigut contactades tinguin un valor de 999, com si fes molt temps que ho van ser.

- **Previous + Poutcome**. L'intuïció ens diria que si una persona ha sigut contactada abans té més probabilitat de contractar el dipòsit, i això és exactament el que veiem. A més, observem que, efectivament, coincideixen els individus que no han sigut contactats amb els que poutcome és unknown. Un altre cop, estan desbalancejades.

- **Variables socioeconòmiques**. El període de temps en el que el conjunt de dades va ser recollit inclou la crisis del 2008. Això afecta molt a aquestes variables, com per exemple euribor3m, que conté variacions molt grans per un període relativament curt. Per a veure la consistència d'aquesta variable, es va consultar aquest índex al període adequat i es va veure que coincideix amb la nostra variable. Un altre exemple de l'efecte de la crisis econòmica és veure com l'índex de confiança dels consumidors es troba en mínims històrics o com el nombre d'empleats decreix.

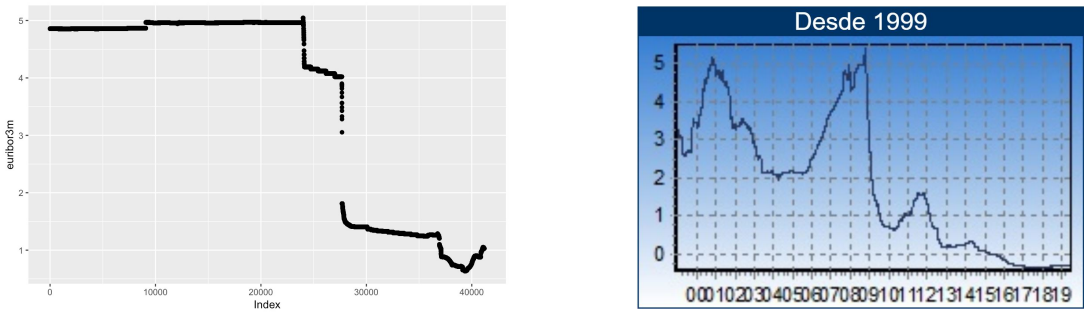

Figure 1: variació euribor3m

3.2 Clustering

Abans de res, volem veure quina naturalesa tenen les nostres dades per tal de veure si tenim diferents tipus de persones que tinguin característiques semblants i es comportin de la mateixa manera a l'hora de contractar un depòsit a llarg termini. D'aquesta manera, podrem esperar una mica el comportament que ens trobarem a l'hora de modelar i quines són les nostres limitacions.

Hem realitzat l'algorismes de clustering k-means. Per aquest hem fet servir el dataset amb les tècniques d'imputació explicades en *apartat 3.2*, ja que no podem tenir cap valor perdut pel models. Pot ser que això introdueixi algun tipus de biaix sobre les dades, però considerem que tenim moltes observacions perquè això sigui menyspreable.

Per trobar el nombre òptim de clústers dins les dades i poder posar en marxa k-means, hem utilitzat l'índex C-H. Hem trobat que aquest assoleix el màxim en k=5 clusters. Ens interessa veure si hi ha patrons que ens puguin predir el que farà una persona quan la truquem. Per això, ens endinsem a veure quines són les proporcions a cada clúster de cada classe.

	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
classe "no"	0	255	4851	4231	6768
classe "yes"	314	1600	759	4186	1295

Els dos primers clústers ens troben dos tipus de persones que són molt probables a que contractin un dipòsit a llarg termini. En concret, dins del primer clúster la proporció de la classe "yes" és del 100%. En el quart clúster, observem que existeix un grup de persones amb característiques semblants que tant poden pertànyer a una classe com a l'altre, per tant, en aquest tenim màxima incertesa. Finalment, en la resta de clusters (tercer i cinquè) trobem grups de persones que majoritàriament ens diran que "no". En definitiva, d'aquesta examinació podem deduïr que la classe "no" és més predible que la classe "yes", ja que té més mostres i ocupen gran part dels clusters sent majoria.

Podem també mirar la relació dels clústers amb les diferents variables. Aquests resultats no són gaire significatiu. El més destacable que hem trobat ha estat que els clusters 1 i 2, que ens contenen bastanta part de la gent que ens contracta un dipòsit, estan molt relacionats amb valors de "$p - days$" < 500, "$previous$" > 1, "$poutcome$" $= success$, $1 <$ "$emp.var.rate$" < 1.3 i $4 <$ "$euribor3m$" < 5. Això ens està dient que si ja he contactat una persona en una altra campanya i aquesta m'ha contractat un producte financier, en aquesta nova campanya tindré forces probabilitats que també ho faci. També, veiem la influència de les variables socioeconòmiques sobre les decisions de la gent.

3.3 Partició de les dades i imputació

Després del pre-procés, ens hem quedat amb un 7.15% d'observacions amb algun valor *missing*, i creiem que és adient no eliminar-los ja que podríem eliminar informació important. En concret, de tots els valors mancants, hi ha un 12.98% de "yes" i un 87.02% de "no", cosa que ens fa veure que no hi ha cap patró en la distribució dels *missings* en el que la variable resposta respecta, ja que segueixen la mateixa proporció que al set de dades original.

Abans d'imputar hem dividit les dades en *train* y *test* (en proporcions de 75% i 25%, respectivament, i de manera que el repartiment de la variable resposta es conservi), ja que imputar totes les dades de forma conjunta i després fer el *splitting* faria que el set de *test* contingués informació del set de *train*, cosa que contradiu el fet que *train* i *test* han de ser independents.

Imputem, doncs, utilitzant la funció *missForest* de la llibreria homònima. *missForest* imputa de forma no paramètrica i serveix per a casos de classes mixtes com és el nostre, basant-se en entrenar un *random forest* a partir de les dades proporcionades per imputar les mancants, conservant interaccions complexes i relacions no lineals.

La imputació, però, té la desavantatge que quan fem l'anàlisi de les dades i creem diferents models a partir d'aquestes estarem carregant una certa incertesa. Aquesta incertesa, però, serà poca ja que disposem de moltes observacions.

3.4 Tractament de les dades (balanceig)

La proporció de les classes original és d'un 11.27% de "yes" i un 88.73% de "no". Aquest desbalanceig fa que els models tinguin un biaix de predicció cap a la classe majoritària. En el nostre cas, és més interessant saber identificar correctament un client que ens vulgui a fer el depòsit, és a dir de la classe minoritària, que un que no ho estigui. Per tant, necessitem balancejar les dades.

Abans de balancejar, sabem que efectuar *Cross Validation* directament sobre les dades amb les proporcions naturals de la població alterades no és correcte, per la qual cosa hem fet una altre partició a les dades amb la qual farem validació simple. Concretament, del conjunt de *train* hem extret un 25% (respecte la mida total, incloent el conjunt de test) de les dades i les hem posat al set de validació. Un cop tenim la divisió de *train*, *validació* i *test* $(50\%, 25\%, 25\%)$, efectuem el balancejat sobre les primeres: les de *train*. Degut a la diversitat de tècniques de balancejat i possibles configuracions, hem creat diferents sets de dades, cadascun balancejat amb diferents proporcions i tècniques. Tots ells comparteixen el mateix set

de *validació* i de *test*.

La tècnica més simple utilitzada ha sigut *undersampling*, que consisteix en agafar només una proporció de les dades de la classe majoritària, les quals hem escollit aleatòriament amb reemplaçament del conjunt de *training*. El nou set de *training*, doncs, té moltes menys dades de les que tenia originalment, cosa que augmenta les probabilitats de fer *overfitting*. Veurem, però, si hi ha un *trade-off* que valgui la pena.

SMOTE, acrònim de *Synthetic Minority Over-sampling Technique*, combina el *oversampling* (US) i l'*undersam-*
-pling de manera que crea dades sintètiques que pertanyen a la classe minoritària a partir de mostres d'aquesta. A més, fa *undersampling*, així doncs agafa menys mostres de la classe minoritària. Tenim, doncs, diferents possibles configuracions ja que es pot tenir la mateixa proporció de dades amb nombre de dades diferents. Amb aquestes dues tècniques de balançejament hem proposat els següents conjunts d'entrenament sobre els quals modelarem:

			nº observacions		
% "yes"	**% "no"**	**Mètode**	**Set 1**	**Set 2**	**Set 3**
50	50	SMOTE	13962	9308	-
		US	4654	-	-
60	40	SMOTE	11635	7679	5757
		US	3874	-	-
70	30	SMOTE	10006	6631	-
		US	3388	-	-
80	20	SMOTE	5930	-	-
		US	2965	-	-

L'avantatge de tenir les mateixes proporcions però menys observacions amb SMOTE pot ser que ens hem inventat un menor nombre de mostres

4 Mètodes i mètriques de validació

Com ja hem mencionat, efectuar *Validació Creuada* amb dades d'entrenament que tenen proporcions de la variable resposta alterades és erroni, degut a que en aquest cas, l'error de C.V. no seria una bona estimació del rendiment d'un model quan l'apliquem a les dades reals, les quals tenen una natura diferent.

Per tant, en lloc d'efectuar *Validació Creuada*, fem validació simple, és a dir, tenim un conjunt de validació independent del d'entrenament i del de test, amb proporcions ja mencionades al document. Després d'entrenar cada model, prediem les dades d'aquest set, i calculem les mètriques de validació corresponents, amb les quals ens basem per tal d'escollir nous hiperparàmetres i per comparar-los entre ells.

Les mètriques de validació són molt importants per escollir els models correctament, ja que depenent de la natura del problema, el significat de millor o pitjor model és variable. La nostra variable resposta és si un client contractarà o no un dipòsit a llarg termini, per tant, hem de considerar si és més interessant predir bé els que sí o els que no o si els dos són igualment importants.

Entrant al context de *bank telemarketing*, encara que per nosaltres, amb la informació que tenim, és molt difícil quantificar els guanys o les pèrdues de la contractació o no d'un dipòsit, sabem que, segons es menciona a [**1**] els serveis de comunicació necessaris, incloent recursos humans i tecnològics, usualment es contracten en *packs* de manera que els costos s'abarateixen. D'altra banda, és fàcil veure que descartar un potencial client no és gens interessant des d'un punt de vista financer, per la qual cosa volem un model que maximitzi els individus que correctament prediu que contractaran el dipòsit condicionat a que

idealment hauríem de contactar a la menor quantitat de gent possible.

Les proporcions que volem maximitzar són, per tant, el True Positive Rate (TPR) que correspon a les prediccions "yes" correctament fetes, i el True Negative Rate (TNR) que correspon al mateix però per les prediccions "no", permetent que aquest últim pugui ser més petit a canvi d'un increment substancial del primer. Per tal de saber quin és el *tradeoff* que més ens interessa mirem la proporció de trucades realitzades que resultaran en el client contractant un dipòsit sobre les trucades totals, és a dir, una mesura d'eficiència. Així doncs, quan parlem de l'eficiència d'un model ens refereirem a aquesta mesura. Aquesta eficiència pot ser la mateixa amb diferents configuracions: potser tenim un TPR baix però un TNR molt alt o tenim un TPR alt amb un TNR molt baix. Per tant, busquem maximitzar aquesta mesura d'eficiència subjecte a que el TPR sigui suficientment gran. Al nostre cas, hem fixat un TPR mínim de 65%, tenint en compte que si trobem diversos models similars en eficiència escollirem com a millor el que tingui un TPR més alt.

Per acabar, remarcar que també hem utilitzat la *F1 score*, però en general la discriminació de models basant-nos en aquesta no era del tot satisfactòria pel nostre criteri.

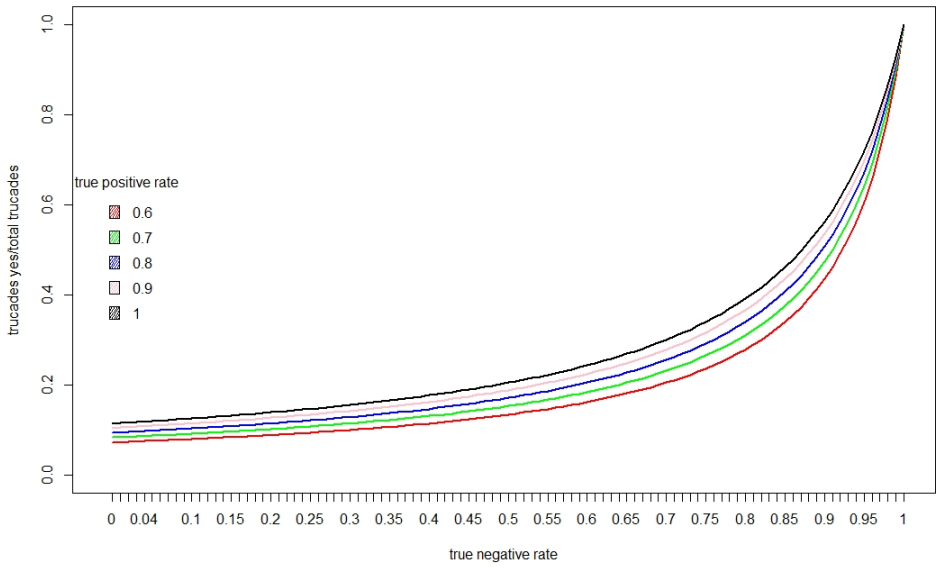

Figure 2: Corba de l'eficiència d'un model en funció del TPR i el TNR.

5 Modelització

La modelització al llarg de tot el procés del treball ha seguit diferents fases. Primer de tot, hem aplicat els models tant lineals com no lineals sobre les dades amb totes les files que contenien algun valor perdut eliminades. Els errors que obteníem eren molt dolents, ja que per exemple, pels mètodes lineals l'error total de *training* i validació no baixava del 20% i els encerts de la classe *yes* eren molt més baixos que els de la classe *no*. El següent pas, ha estat treballar amb totes les 40000 files del conjunt de dades imputades seguint els mètodes explicats en l'*apartat 3.2*. D'aquesta manera aconseguíem disminuir l'error total fins arribar en alguns casos al 10% però les diferències entre les prediccions de les dues classes seguien sent presents. Per això i, tal com ja s'ha explicat en l'anterior *apartat 3.3*, hem tornat a modelar amb diferents conjunts de dades en els quals les proporcions de les dues classes i els mètodes de balanceig són diferents.

Així hem aconseguit millorar les prediccions de la classe minoritari, a canvi d'augmentar en certa manera l'error total.

Per cada mètode, hem creat una taula on es representen els resultats obtinguts: el percentatge d'error total sobre el conjunt de validació, el percentatge d'encerts de la classe "yes", el percentatge d'encerts de la classe "no" i l'eficàcia del model (si trobem que és NA és perquè el model no ens compleix la condició que el TPR ha de ser superior a 0.6). Pel mètodes amb hiperparàmetres, també hem creat una columna amb el millor que obtenim a partir de validació. En la primera fila sempre hi ha els resultats amb el conjunt de dades imputat però sense balancejar (l'hem anomenat *Original* perquè conserva les proporcions reals de les dues classes) i en la segona fila el millor model entrenat amb un dels conjunts de dades balancejats (s'indica explícitament quin és) i seleccionat a partir de les tècniques explicades a *l'apartat 4*.

5.1 Anàlisi discriminant lineal (LDA) i anàlisi discriminant quadràtic (QDA)

El millor classificador possible que podem fer sobre les nostres dades és el classificador Bayesià. Desafortunadament, no el podem aplicar directament perquè no sabem els paràmetres reals de la població ni si la distribució de les dades és una Gaussiana. Per tant, aplicarem els mètodes LDA i QDA com una aproximació a aquest. Amb l'anàlisi discriminant lineal assumim que les dades de totes les classes tenen una matriu de covariàncies comuna, mentre que l'anàlisi discriminant quadràtic suposa que cada classe té la seva pròpia matriu de covariàncies.

A partir dels resultats obtinguts abans del balanceig de les dades, comprovem com l'error total que aconseguim en ambdós mètodes pràcticament no passen del 10%. Malgrat això, les prediccions de la classe minoritària són relativament baixes comparades amb la de la classe majoritària, especialment en LDA. Sorprenentment, els resultats obtinguts amb QDA són millors del que ens podríem haver esperat respecte la classe "yes", superant lleugerament el 50% d'encerts.

Un cop balancejades les dades, hem notat que el rang de valors d'errors i encerts en els quals varien els diferents *datasets* són molt semblants. Els dos conjunts de dades que ens aconsegueixen millors resultats són els que estan balancejats en proporcions 60-40, tant SMOTE com US. Ara, els resultats del mètode LDA sempre són millors que amb QDA i ens assoleix proporcions molt bones, arribant a predir per igual les dues classes. També hem observat com els resultats del QDA no varien gaire respecte els resultats sense balanceig per tots els *datasets* balancejats. Per tant, per aquest mètode no aconseguim cap millora en la predicció de la classe "yes", que és el nostre objectiu, i és doncs un model que no ens permetrà generalitzar bé la població.

A partir de les funcions discriminants que ens creen els models, podem apreciar quines són les variables que més ens influencien la classificació. Per el nostre conjunt de dades, les dues variables amb més influència són "emp.var.rate" i "cons.price.idx", totes dues variables socioeconòmiques.

		Error VA	TPR	TNR	Eficiència
Original	LDA	10.91	38.96	95.54	NA
	QDA	13.26	53.79	90.80	NA
60-40 (US)	LDA	27.49	72.21	72.54	25.27
	QDA	13.83	55.70	90.09	NA

5.2 Regressió multinomial

La regressió logística és un tipus d'anàlisi de regressió utilitzat per predir el resultat d'una variable categòrica en funció de les variables predictores. Aquesta tècnica pertany als anomenats *Models Lineals*

Generalitzats (GLM), en particular usant com a funció d'enllaç el *logit*. Els resultats que hem obtingut han de ser el mateixos que si modelèssim una xarxa neuronal MLP d'una sola capa.

Un cop modelada la regressió logística sobre les dades de training, podem aplicar la funció "step" de R que ens simplifica el model eliminant les variables menys importants progressivament fins que s'obté el model amb menor AIC possible. Amb aquest procediment, a més, aconseguim saber quines són les variables més significatives. En el nostre cas hem obtingut les següents: "contact", "age", "euribor3m", "emp.var.rate", "nr.employed", "poutcome" i "month", entre les quals tornen a destacar les variables socioeconòmiques.

En aquesta tècnica, també tenim un hiperparàmetre: el llindar de classificació (P), que ens indica a partir de quina probabilitat es classifica una observació com una classe o altre. Aquest ens aporta molta flexibilitat a l'hora de modelar. Per exemple, encara que el conjunt de dades no estigui balancejat podem disminuir tant com es vulgui aquest llindar i així aconseguir millors prediccions de la classe minoritària. En general, notem que amb valors de P molt baixos aconseguim que el model ens predigui mllor la classe "yes" (errors de VA grans) i amb valors de P grans l'altre (errors de VA petits). Per tal de triar quin és el valor de P que millor generalitza les nostres dades, hem provat a modelar per un seguit de probabilitats (diferents segons el dataset) i hem seleccionat aquella en que aconseguiem un millor "tradeoff" entre TPR i TNR.

Tot i així, els valors obtinguts en les dades originals no són del tot satisfactoris i, per tant, hem modelat sobre dades balancejades jugant amb l'hiperparàmetre. En general, els resultats assolits amb tots els conjunts de dades balancejats són satisfactoris, assolint entre 60% i 70% per les dues classes. El millor model possible l'hem obtingut modelant un conjunt de *training* en el qual les noves proporcions són molt més grans per la classe minoritària original. A més, hem comprovat que aquest coincideix amb el model de menor AIC entre tots els modelats. En concret, assoleix un AIC de 2397.6, ja que és el mètode amb que aconseguim fer un *feature selection* més alt.

	Llindar	Error VA	TPR	TNR	Eficiència
Original	0.1	20.03	64.88	81.92	NA
80-20 (US)	0.7	29.20	74.73	70.31	24.45

5.3 KNN

KNN és un algorisme de classificació que permet predir la classe d'una observació com la classe majoritària entre els k veïns més propers de les dades de training. Aquest mètode no crea cap model, sinó que ho són les pròpies dades.
Hem considerat dos casos a l'hora de modelitzar l'algorisme: primer amb les dades original i després hem estandarditzat les variables contínues perquè sabem que KNN sol funcionar una mica millor.

Primer de tot, ha calgut binaritzar les variables categòriques en *dummies* perquè la funció *knn* calcula internament distàncies euclidianes entre els punts. Per això, hem creat *nleves*-1 variables noves per cada variables categòrica de *nlevels* factors. A continuació, hem hagut de triar l'hiperparàmetre k que ens diu en nombre de veïns que cal mirar per classificar una dada. Per això, hem calculat els errors de validació per un les k de 1 fins a $\sqrt{N}$ (com a convenció) de tres en tres, sent N la quantitat d'obervacions del conjunt d'entrenament.

Pels dos casos proposats comprovem com els resultats són molt semblants, però sempre són una mica superiors en els models creats a partir de les dades estandarditzades. En concret, el millor model que hem trobat pels dos fa servir l'hiperparàmetre k=28 i, com hem dit, el de les dades estandaritzades ens prediu millor la classe "yes".

	k-veïns	Error VA	TPR	TNR	Eficiència
Original	7	10.77	26.43	97.30	NA
60-40 (US)	28	31.10	74.51	68.18	23.14

5.4 Naive Bayes

Naive Bayes és un mètode de classificació probabilístic que utilitza les probabilitats a posteriori de pertànyer a les classes de la variable resposta per a predir. Assumeix que tots els factors del conjunt de dades són independents.

Per tal de realitzar aquest mètode, només ens cal ajustar les dades de training i ja se'ns calculen totes les probabilitat a posteriori que necessitem. Les mesures que obtenim amb tots els models són molt semblants, superant el 65% per la classe minoritària i el 70% per la majoritària. Aquests resultats són molt millors del que podríem esperar, ja que les suposicions que aplica aquest algorisme solen ser bastant diferents de la realitat.

	Error VA	TPR	TNR	Eficiència
Original	15.77	53.37	88.20	NA
80-20 (US)	26.57	70.74	73.76	25.74

5.5 MLP de dues capes

Modelem ara un Multi-Layer Perceptron de dues capes, és a dir, una oculta i l'altra de sortida amb la funció *nnet* de la llibreria *nnet*, amb la qual podem ajustar tant el paràmetre de regularització de la funció d'error (*decay*) com el número de neurones de la capa oculta (*size*). Un MLP té la capacitat d'aprendre relacions complexes no-lineals que les dades puguin presentar amb un número raonable de neurones a la/les capa/es oculta/es, i quan no té capa oculta és equivalent a un model de regressió logística. Com el problema MLP no és convex, presenta mínims locals, als quals convergirem depenent dels pessos inicials. Primer, fixem el número de neurones de la capa oculta seguint la mateixa heurística utilitzada a **[1]** a $round(\frac{M}{2} = 9)$, on M és el número de *features* d'entrada de la xarxa. Seguidament, per a tots els conjunts de dades balancejats i estandarditzats a mitjana 0 i desviació típica 1, busquem el paràmetre de regularització més adient d'entre el conjunt {0.001, 0.00398, 0.01585, 0.06309, 0.2512, 1}. D'aquesta cerca trobem que la xarxa més adient és formada per 9 neurones a la capa oculta, $decay = 0.06309$ i entrenada amb el conjunt de training US 60-40. Al provar amb 7 neurones a la capa oculta i amb el mateixos paràmetres de regularització, però, trobem un model millor en el cas del mateix *decay* que abans, 0.06309. Amb aquesta configuració, provem diferents inicialitzacions, però cap és millor, i per tant, ens quedem amb els resultats següents:

	Error VA	TPR	TNR	Eficiència
Original	10.88	26.51	97.17	NA
60-40 (US)	31.39	71.1	68.29	22.37

El qual és un *tradeoff* bastant balancejat. Hem observat que quan més gran es la proporció de "yes" al conjunt d'entrenament, més alt es el TPR a canvi d'un TFR molt baix i viceversa. Aquest comportament era d'esperar, ja que es la raó per la qual hem decidit utilitzar dades balancejades.

5.6 SVM

SVM és un mètode de separació lineal de dades basat en construir un hiperplà que minimitza l'error de classificació. Aquest pot separar les dades al espai original o a un *feature space* a on, amb sort, la separació serà millor. Les operacions al *feature space* es fan de manera implícita amb la funció de kernel,

de la qual hi han diverses bones opcions. Nosaltres utilitzem la funció *ksvm* de la llibreria *kernlab*. Amb tots els conjunts de dades estandarditzades, doncs, provem tres funcions de kernel per cadascun: lineal, Gaussià RBF i polinòmica de grau 2. Cada un d'aquest kernels té diferents hiperparàmetres per ajustar, encara que tots tenen en comú el paràmetre de regularització dels errors (C).

1. Kernel lineal: només hem d'ajustar el paràmetre C. Provem pels valors {0.01, 0.1, 1,10, 100}
2. Kernel polinòmic: fixem *l'offset* o biaix a 1, i provem C pels valors {0.01, 0.1, 1,10, 100}.
3. Kernel RBF Gaussià: A més de C, que provem pels valors {0.01, 0.1, 1,10, 100}, podem ajustar l'hiperparàmetre *sigma*, que posem com "automatic" de tal manera que es calcula un bon valor per a les nostres dades utilitzant una heurística i que resulta ser bastant adient.

El fet que haguem fixat hiperparàmetres com *l'offset* o *sigma* és degut a la quantitat de conjunts d'entrenament que tenim i el llarg temps de computació que ens presentaria fer *Grid Search* de la millor configuració. Com els resultats obtinguts, però, són molt similars als millors obtinguts amb tots els altres mètodes, la millora que suposaria buscar per molt valors *d'offset* o de *sigma* no seria significativa, si hi hagués.
Els millors resultats respecte el nostre criteri de validació són els següents.

	Error VA	TPR	TNR	Eficiència	Kernel	C
70-30 (SMOTE)	28.76	72.63	71.054	24.39	RBF Gaussià (Sigma auto.)	1
70-30 (SMOTE)	34.83	76.21	63.74	21.27	Lineal	100
60-40 (US)	26.1	71.27	74.24	26.24	Quadràtic (Offset = 1)	0.01

Observem com els tres resultats són bastant positius amb valors de C diferents, remarcant la importància de la regularització dels models. Amb el kernel lineal és amb el que hem aconseguit més TPR, que ens interessa, però a la vegada és menys eficient que els altres dos. Descartem el kernel quadràtic perquè és el que menys TPR té i només és un 1.85% més eficient que el RBF Gaussià, i ens quedem amb el model del kernel lineal, que té un 3.58% més de TPR amb una eficiència un 3.12% menor comparat amb el SVM amb kernel RBF Gaussià.

5.7 Random Forest

Un Random Forest és un *ensemble* d'arbres de decisió independents els uns dels altres els quals han sigut entrenats individualment de conjunts de *bootstrap* diferents (amb igualtat de distribució) i que "voten" la classe de l'observació a predir. En el cas de classificació binària, es vota per majoria (50% + 1). Cada arbre de decisió discrimina utilitzant una conjunt aleatori de m predictors extret del *set* de dades d'entrenament. A més, aquest mètode retorna una estimació fiable de l'error generalitzat del model: Out Of the Bag error (OOB). No ens basem en aquest per escollir el millor model, però, perquè no pren en compte el desbalanceig de les dades.

Quan entrenem un model amb Random Forests, podem especificar el nombre d'arbres que formaran *l'ensemble* i el número m de variables de cada arbre. Entrenem, doncs, un Random Forest per a cadascun dels nostres conjunts d'entrenament amb la funció *RandomForest* de la llibreria homònima.
Aquest mètode no és computacionalment molt demandant, per la qual cosa hem fixat 500 arbres per Random Forest i, amb tots els conjunts de dades, mitjançant una gràfica, hem vist que amb 200 ja eren suficients, ja que en tots els casos s'havien estabilitzat els errors (figura 3).

El número de variables m per defecte al nostre cas és 4, per la qual cosa només pels models que millors resultats havien aconseguit a la fase anterior hem provat a variar el valor de m de 2 a 15, d'on observem que passar d'utilitzar de 2 a 3 variables dóna resultats lleugerament diferents però que utilitzar més variables altera el comportament del model de forma negligible. El resultat més satisfactori ha sigut:

	Error VA	TPR	TNR	Eficiència	OOB	m
ORIGINAL	10.42	29.75	97.27	NA	11.04	6
60-40 (US)	29.92	74.94	69.45	23.98	27.65	3

Per tant, d'aquest apartat concloem que la millor configuració de RandomForest ha sigut amb 200 arbres, 3 variables cadascun. Observem com, efectivament, no ens podíem basar en el OOB per escollir el millor model.

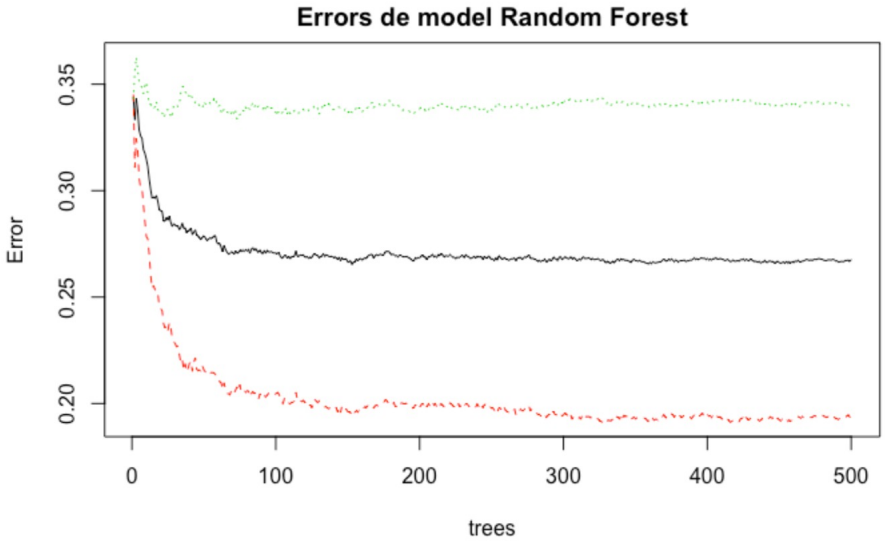

Figure 3: Veiem com els errors s'estabilitzen

6 Comparació de resultats i conclusions

Després d'analitzar i entendre les dades, visualitzar relacions, pre-processar, balancejar i provar diferents tècniques de modelatge, tant lineals com no lineals, hem aconseguit models de predicció que ens aconsegueixen generalitzar força bé la població de les dades. En totes les tècniques que hem aplicat hem optat per models amb una alta sensitivitat (True positive rate) amb un cert equilibri també amb valors alts de True Negative Rate, ja que així aconseguim arribar al màxim nombre de persones que ens contractaran el dipòsit i al mateix temps no gastem molts recursos en trucades innecessàries, personal...

Així doncs, només ens queda seleccionar algun model d'entre els millors de cada mètode d'aprenentatge automàtic. A continuació, els representarem visualment en una gràfica amb totes les corresponents mesures que hem anat calculant (errors validació, true positive rate, true negative rate i eficiència).

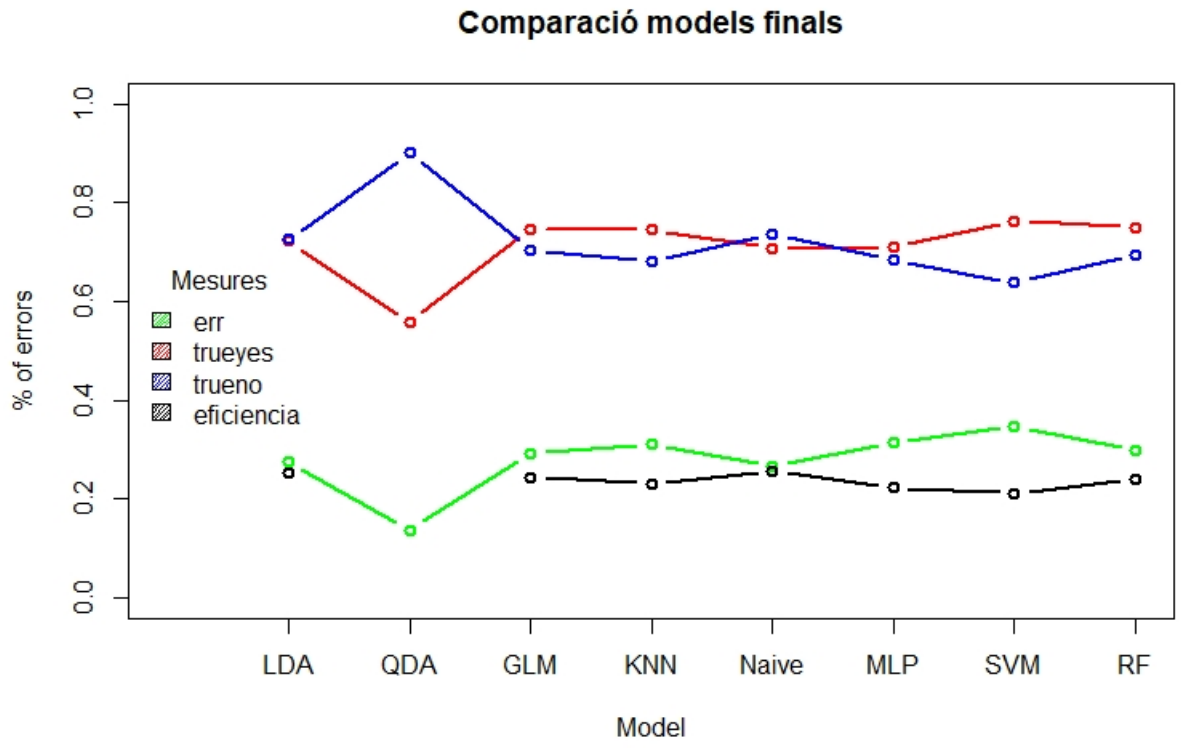

Figure 4: Comparació models

Seguint els nostres dos pricipals objectius (maximitxar TPR i eficiència) veiem que hi ha dos models que s'assemblen molt i aconseguixen gairebé els mateixos valors per cada mesura. Aquests dos assoleixen un percentatge de TPR del voltant de 75% i una eficiència del 24%, i corresponen a la regressió logística i al Random Forest.

L'últim pas és veure com de bé aquests models generalitzen les característiques de la població, aplicant-los sobre el conjunt de test. Els resultats obtinguts han estat els següents:

	Error TE	TPR	TNR	Eficiència
Regressió logística	0.2985336	0.7385965	0.6968439	0.3539302
Random forest	0.3081480	0.7438596	0.6853773	0.3483261

Veiem com clarament els resultats són semblants, tal i com ja veiem en els errors de validació.

En conclusió, hem obtingut dos models, un de lineal i l'altre de no lineal, ambdós amb un error de predicció total del 30%. A més, aconseguim arribar a gairebé el 75% dels potencials clients contactant a aproximadament només un 35%. També estem minimitzant les trucades que faig en total, aconseguint així que un un 35% del total de trucades acabin en la contractació del depòsit.

Per acabar, volem fer un seguit de reflexions sobre el procés del treball i la tria dels models. Primer de tot, cal tenir molt clar des del principi quins són els objectius que jo em proposo a l'hora de començar a modelar. En aquest cas, volem obtenir el màxim nombre de gent que contracti el dipòsit al banc. Però això no és l'únic a tenir en compte, ja que es pot trucar a tots els clients i aconseguir que el 100% de persones de la classe "yes" mel contractin. Per tant, cal tenir en compte què ens importa més: perdre

un client disposat a contractar el dipòsit o el cost de recursos humans i trucades. En el nostre cas, aquests han estat presents sempre, però ens hem vist amb el problema que a vegades la tria era una mica subjectiva al punt de vista del programador.

Hem comprovat com la població que se'ns presenta en les dades és difícilment predictible, ja que en voler augmentar el true positive rate estem també augmentant l'error total fin un 30%. Hem trobat que en la majoria dels casos, són les variables socioeconòmiques les que ens ajuden a decidir i que segurament, amb variables predictores que no s'han inclòs al conjunt de dades que hem treballat per questions de protecció de dades el model final es podria millorar.
Finalment, voldríem comentar que hem aprés moltes coses com tècniques de balancejat de dades, d'avaluació de models, d'imputació, maneres d'interpretar resultats per un problema en concret, entre d'altres, i que com a continuació d'aquest projecte, es podrien expandir les tècniques de selecció de models, és a dir, utilitzar mètriques com l'AUC o ALIFT (en aquest context, ja que és màrketing) i ampliar una mica a l'apartar de *feature selection* a partir dels arbres aleatoris o de la xarxa neuronal.

7 Bibliografia

[1] 2014, Sergio Moro, Paulo Cortez, Paulo Rita: A data-driven approach to predict the success of bank telemarketing.
[2] 2011 SMOTE: Synthetic Minority Over-sampling Technique