MEMÒRIA

KAGGLE - APC

CLASSIFICACIÓ

Universitat Autònoma de Barcelona

Oriol Camps Isús, 1558778

INTRODUCCIÓ

Link al Kaggle del projecte

Context:

En aquest kaggle tenim un dataset de tipus tabular. El conjunt de dades prové de la U.S. Small Business Administration (SBA).

L'SBA dels Estats Units es va fundar l'any 1953 amb el principi de promoure i ajudar les petites empreses al mercat de crèdit dels EUA. Les petites empreses han estat una font principal de creació d'ocupació als Estats Units; per tant, fomentar la formació i el creixement de petites empreses té beneficis socials en crear oportunitats laborals i reduir l'atur.

Hi ha hagut moltes històries d'èxit de noves empreses que reben garanties de préstecs de la SBA, com ara FedEx i Apple Computer. Tanmateix, també hi ha hagut històries de petites empreses i/o empreses emergents que no han complert amb el pagament dels seus préstecs garantits per la SBA.

Anàlisis del data set:

Tenim un dataset amb 899164 dades i 27 atributs, hi ha 10 atributs de tipus numèric, 5 de tipus categòric i la resta de tipus object, poden ser dates, strings, etc.

El nostre atribut objectiu seria **MIS_Status** el qual et diu si un préstec ha sigut aprovat o no.

Descripció d'atributs:

En aquesta taula es poden veure tots els atributs amb una petita descripció de cada un i el tipus de variable que representen:

Nom	Descripció	Tipus
LoanNr_ChkDgt	Identifier Primary key	int
Name	Borrower name	str
City	Borrower city	str
State	Borrower state	str
Zip	Borrower zip code	int
Bank	Bank name	str
BankState	Bank state	str
NAICS	North American industry classification system code	int
ApprovalDate	Date SBA commitment issued	data
ApprovalFY	Fiscal year of commitment	data
Term	Loan term in months	int
NoEmp	Number of business employees	int
NewExist	1 = Existing business, 2 = New business	category
CreateJob	Number of jobs created	int
RetainedJob	Number of jobs retained	int
FranchiseCode	Franchise code, (0 or 1) = No franchise	int
UrbanRural	1 = Urban, 2 = rural, 0 = undefined	category
RevLineCr	Revolving line of credit: Y = Yes, N = No	booleano
LowDoc	LowDoc Loan Program: Y = Yes, N = No	booleano
ChgOffDate	The date when a loan is declared to be in default	data
DisbursementDate	Disbursement date	data
DisbursementGross	Amount disbursed	int
BalanceGross	Gross amount outstanding	int
MIS_Status	Loan status 0 = CHGOFF, 1 = PIF	booleano
ChgOffPrinGr	Charged-off amount	int
GrAppv	Gross amount of loan approved by bank	int
SBA_Appv	SBA's guaranteed amount of approved loan	int

Es pot veure com hi ha atributs de tots tipus que haurem de tractar per tal que els models siguin els mes eficients possibles.

PREPROCESAMENT

Per a processar les dades, el primer que he fet ha sigut eliminar les columnes amb identificador únics ja que no aporten informació útil per a predir l'objectiu.

Seguidament he mirat els valors Nans de tots els meus atributs, de primeres veig que hi ha una gran quantitat de nan en diferents columnes, començant per ordre, he vist les columnes de *City* i *State* aquestes tenen 30 i 14 respectivament, per tant he eliminat les files directament, ja que tinc una gran quantitat de files i la pèrdua de informació no és gran. A continuació, he tractat la meva variable objectiu, ja que aquesta també tenia valors nans, aquests també els he eliminat directament ja que fer suposicions sobre aquesta columna podria afectar molt a l'hora de predir valors reals, a més de tractar els nans, he canviat els valors a 1 i 0 per a facilitar tant la visualització com el model.

He vist que hi ha columnes les quals tracta com a object però que en realitat son floats, així que he fet el canvi de tipus a aquestes columnes.

Per a tractar els nans de l'atribut *NewExist,* el qual diu si la empres és de nova creació, he fet el supòsit que si una empresa te retenció de treballadors, és a dir que l'atribut *RetainedJob* és superior a 1, significa que no és de nova creació, per lo tant se li assigna un 1, així he passa de tenir 136 valors nan a tenir nomes 16 valors.

He eliminat columnes les quals donaven informació posterior a l'aprovació d'un préstec per tant no s'hauria de tenir en compte per a predir l'objectiu.

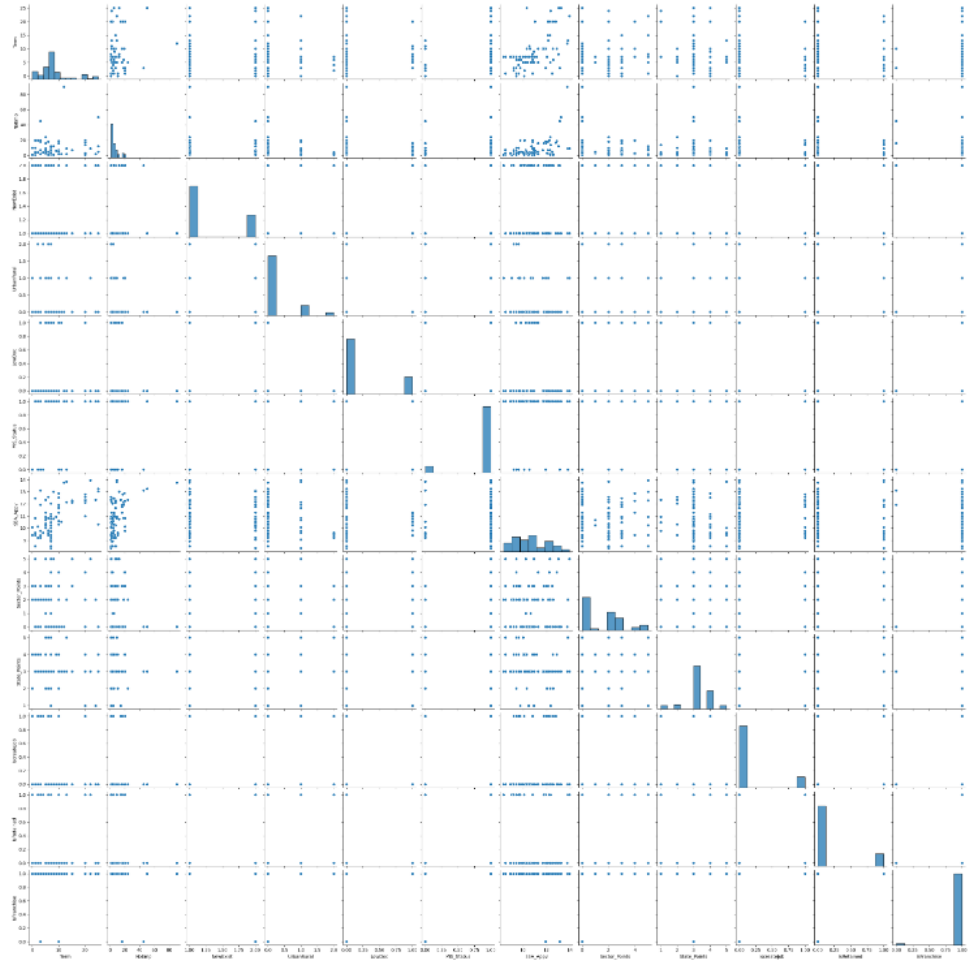
En aquest punt només hem quedaven valors null a les columnes *RevLineCr* i *LowDoc*. He començat tractant els valors de *LowDoc,* primer guardem totes les files amb valor 1, en aquestes files he buscat un patró amb altres columnes, he vist que mes del 75% de les nostres que tenien un valor a la columna *GrAppv* més petit que 100.000 i mes petit que 93 a la columna *Term* tenien el valor 1 a la columna objectiu, per tant, en les files que complien amb aquestes condicions he assignat un valor 1, en aquest cas eren 1565 files, he repetit el procés per el valor 0, la resta de files les he eliminat. He intentat repetir el procés amb l'atribut *RevLineCr* però no he trobat ningun patró, per lo tant l'elimino, també elimino les columnes que representes dates ja que tenen molta distribució de valors.

La columna *NAICS* els valors importants son els dos primers dígits, per tant elimino la resta del numero, per tal de classificar tant els valors de *State* i el mateix *NAICS* he assignat uns punts a aquests valors en funció del seu ratio de fraus i així reduir el nombre de classes que té cada atribut, estan distribuïts tal que en quant menys ratio més punts amb punts de l'1 al 5. Un cop assignats els punts eliminem les columnes.

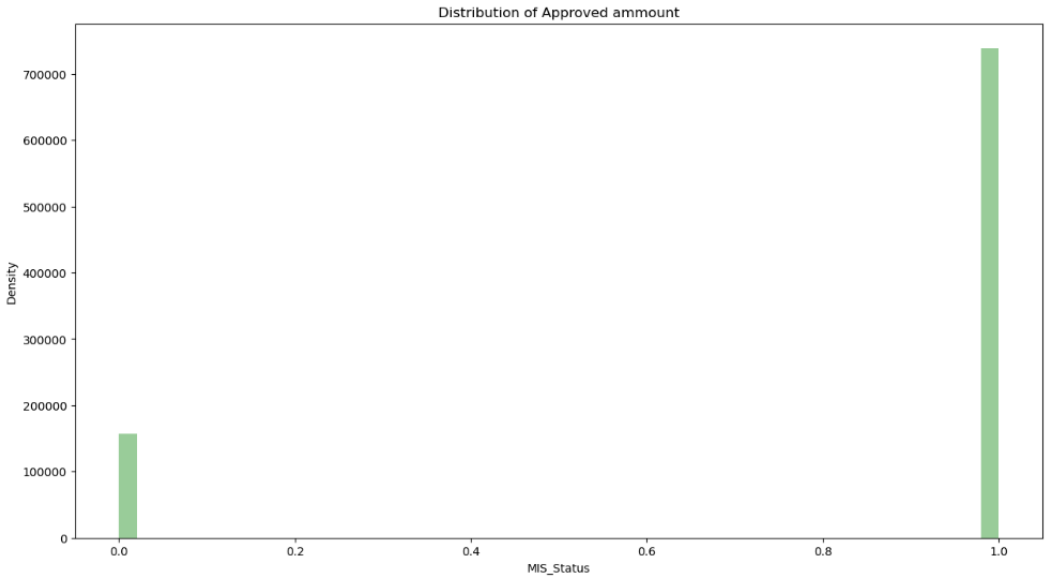
Per últim canviem les columnes booleanes de True i False a 1 i 0. També he normalitzat els valors de la variable utilitzat el logaritme.

Per lo tant, en el dataset final ens queden 896231 dades i 12 atributs.

He fet pairplot per a veure tant la distribució de les dades com la seva relació.



A més també he fet un histograma de la variable objectiu per a veure la distribució de classes.



Com es pot veure hi ha una gran diferència de mostres entre les dues classes, hi ha unes 7 vegades mes 1 que 0, aquest problema es tracta utilitzant certs paràmetres en els models, més endavant s'explica.

METODES UTILITZATS i AVALUACIÓ DE MODELS

Previ a la utilització de models, és la partició del dataset, en el meu cas he escollit dividir en train i test ja que amb aquesta partició ja puc fer-me una idea de quin és el millor classificador, a més que amb la quantitat de dades que tinc el entrenament dels models seria òptim.

Per a fer la comparativa de models m'he fixat tant en al confusion matrix com en les mètriques.

Per tal de solucionar el problema del desbalanceig de les dades objectiu he decidit utilitzar un model el qual et permet balancejar les dades de forma que quedin ben repartides, aquest model és el EditedNearestNeighbours de la llibreria imbalanced, aquest et permet mantenir les dades de la classe majoritària que son més properes a les classes minoritàries. En el següents gràfics es veu el canvi en la distribució de les classes abans i després d'aplicar el model.

Per a fer prediccions i avaluar el data set, he fet una tria dels millors classificadors, al final he utilitzat:

Random Forest

Per a buscar els millors híper paràmetres per el model he utilitzat la força bruta, he buscat els paràmetres que més podien influir en el resultat del model i he trobat que per aquest cas el més rellevant era el criteri d'split del model, en aquest cas els possibles valors serien 2, gini i entropy, entrenant els models amb els diferents criteris veiem que el millor es entropy

Linear Suport Vector Clasifier (Linear SVC)

En quant als paràmetres d'aquest model, els més rellevants son el penalty i el loss ja que son els que més defineixen l'eficiència del model, el paràmetre penalty pot tenir els valors l1 i l2, el loss pot tenir hinge i squared_hinge, per a testejar quin és el millor he buscat entre totes les combinacions possibles entre aquests dos paràmetres i he trobat que la millor és l2 i squared_hinge.

Decision Tree

Els paràmetres d'aquest model son molt similars als del random forest, en aquest cas també tenim criteri d'split el qual pot tenir els mateixos valors que en el cas anterior, al igual que abans, el millor ha sigut gini.

KNeighbours

En aquest cas el paràmetre més rellevant es la k, per a esbrinar quina k és la millor he escollit una sèrie de valors per la k els quals van augmentant i posteriorment he fet una gràfica per a veure on estava el màxim, així he trobat que la millor k és 2, ja que com es veu a la gràfica, en quant la k augmenta el accuracy disminueix.

Regressió Logística

Els paràmetres del model son molt variats, però els més importants son el penalty i el solver, tots dos modifiquen la manera de fer prediccions que te el model, en aquest cas penalty tenia els valors l1, l2 i elasticnet, en canvi solver podia ser lbfgs, liblinear, sag i saga, a continuació he seguit amb la mateixa estratègia, he entrenat totes les combinacions possibles i he mirat quina és la millor, en aquest cas he trobat que assignant l1 a penalty i liblinear a solver obtenia la millor puntuació.

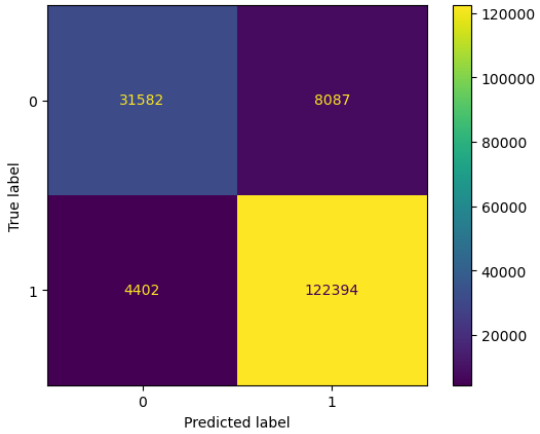
Xarxa Neuronal

En aquest cas la modificació de paràmetres és mes complexa, però al final he vist que les diferències en l'eficàcia del model amb els diferents paràmetres no era molt gran, tot i així hi ha un que si és rellevant, a l'hora de compilar és important que el paràmetre loss es tingui en compte ja que és allò en que es basa el model per a saber si esta millorant o no, per això he escollit CategoricalCrossentropy ja que és el més adequat per a classificació.

Resultats

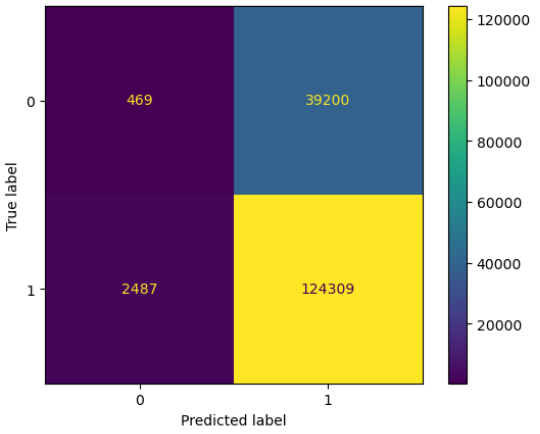
Un cop optimitzats els paràmetres de cada model, s'ha de fer la comparativa entre ells per a veure quin és el millor, per a avaluar els models hem fixaré en la confusion matrix i en el accuracy, he escollit aquests dos ja que amb la confusion matrix puc veure que tant be fa les prediccions i si hi ha algun error de overfiting o amb el des balanç de dades. Els resultats dels diferents models han sigut aquests:

RANDOM FOREST



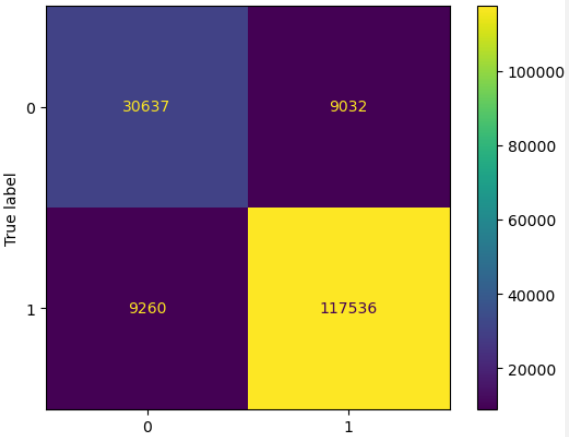
	precision	recall	f1-score	support
0	0.88	0.80	0.83	39669
1	0.94	0.97	0.95	126796
accuracy			0.92	166465
macro avg	0.91	0.88	0.89	166465
weighted avg	0.92	0.92	0.92	166465

LINEAR SVC



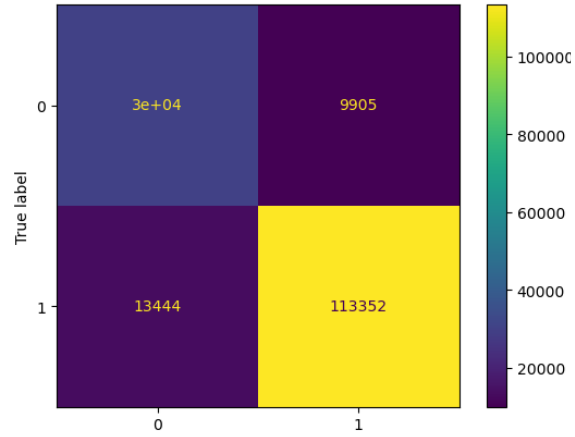
	precision	recall	f1-score	support
0	0.16	0.01	0.02	39669
1	0.76	0.98	0.86	126796
accuracy			0.75	166465
macro avg	0.46	0.50	0.44	166465
weighted avg	0.62	0.75	0.66	166465

DECISION TREE



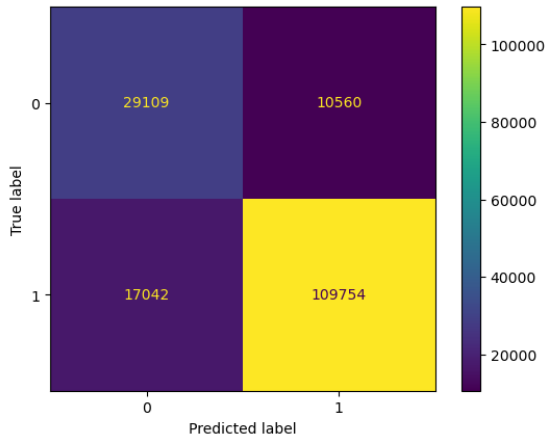
	precision	recall	f1-score	support
0	0.77	0.77	0.77	39669
1	0.93	0.93	0.93	126796
accuracy			0.89	166465
macro avg	0.85	0.85	0.85	166465
weighted avg	0.89	0.89	0.89	166465

KNN



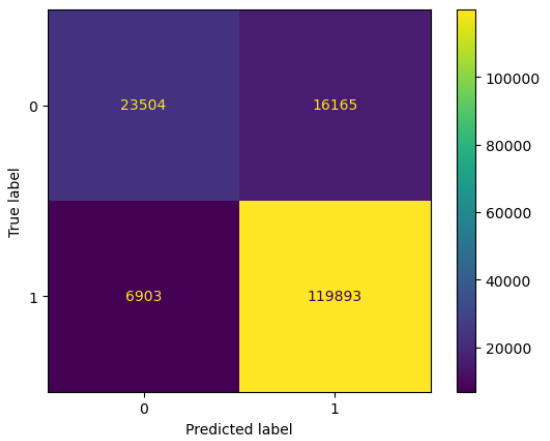
	precision	recall	f1-score	support
0	0.69	0.75	0.72	39669
1	0.92	0.89	0.91	126796
accuracy			0.86	166465
macro avg	0.80	0.82	0.81	166465
weighted avg	0.86	0.86	0.86	166465

RED NEURONAL



	precision	recall	f1-score	support
0	0.63	0.73	0.68	39669
1	0.91	0.87	0.89	126796
accuracy			0.83	166465
macro avg	0.77	0.80	0.78	166465
weighted avg	0.85	0.83	0.84	166465

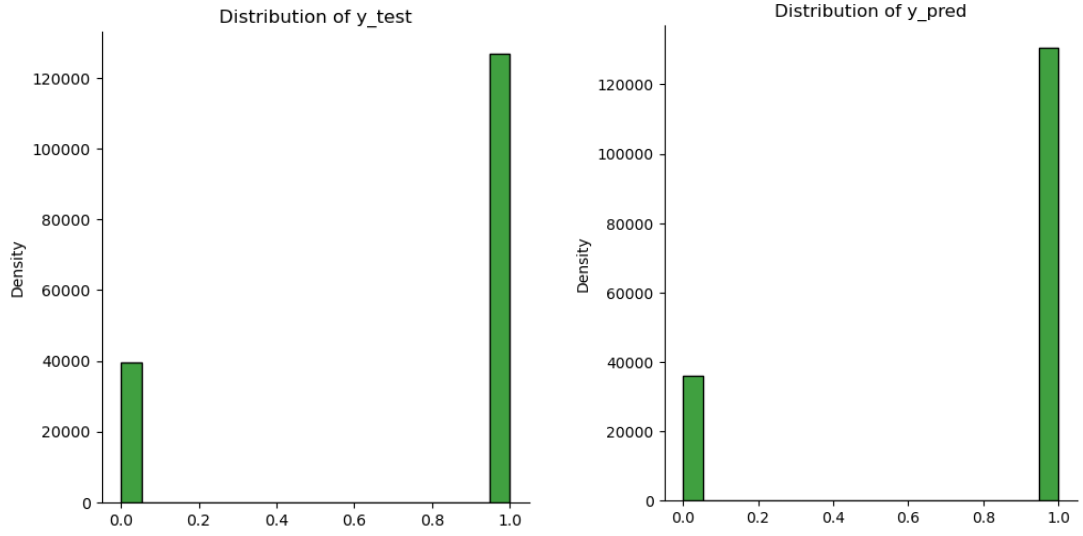
Regresio Logistica



	precision	recall	f1-score	support
0	0.77	0.59	0.67	39669
1	0.88	0.95	0.91	126796
accuracy			0.86	166465
macro avg	0.83	0.77	0.79	166465
weighted avg	0.86	0.86	0.85	166465

Com es pot veure, el millor model és el Random Forest ja que el accuracy es el mes alt i gràcies a la confusion matrix es veu com no hi ha undersampling, com per exemple li passa al LinearSVC, aquest model simplement diu que sempre s'ha de donar el préstec llavors no te un alt nivell de predicció.

També he fet dos histogrames, un que et mostra la distribució de dades del conjunt a predir i un altre dels resultats fets pel Random Forest.

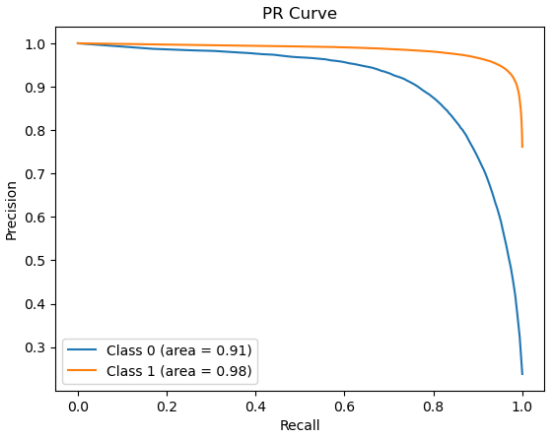


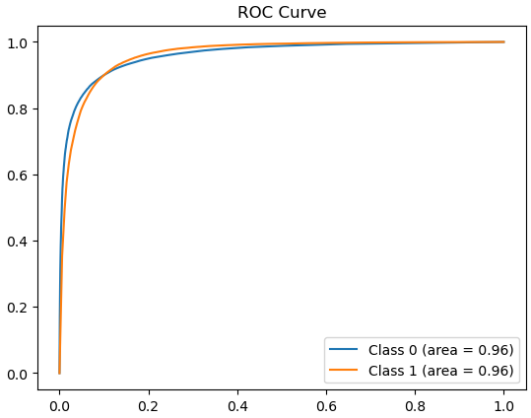
També he fet el càlcul de la proporció de 0 i 1 que hi ha entre el real i el predit

```
                0    1
% of predict:  0.22 0.78
% of test:     0.24 0.76
```

Com es veu la distribució és similar en els dos casos i per tant no hi ha possibilitat de undersampling de classes.

A mes, he fet les curves roc i precision-recall per a assegurar la fiabilitat del model, ja que son una font d'informació que ajuda a verificar els resultats obtinguts en els models i a la vegada he calculat el AUC de cada corba (es mostra a la llegenda dels gràfics).





En els dos casos la puntuació es favorable, en quant a la precission-recall es veu com la classe 0 és més difícil de predir, això pot ser degut a la falta de dades en comparació amb la classe 1, tot i així es una bona corba.

CONCLUSIONS

El data set donat per aquest cas és molt extens ja que conta amb casi 900.000 mostres, a més que hi ha una gran quantitat d'atributs que donen informació innecessària, tot això fa que la neteja del data set sigui més complicada ja que al tenir tanta quantitat de mostres els errors es disparen i la neteja d'aquests pot ser decisiva per el correcte entrenament dels diferents models. Per altre banda, aquest data set, al ser tant extens, et permet fer diferents models per a predir altres atributs com, per exemple, un cop decidit si donar el préstec o no també has de predir la quantitat que dones a les diferents empreses.

També s'ha vist que és molt importat el tractament de undersampling de la variable objectiu ja que sense ell els models podrien estar fent prediccions errònies tot el temps ja que simplement dirien que si a tot, hi ha altres maneres de tractar aquest problema però aquesta he cregut que seria la més eficient degut a la gran quantitat de dades que hi ha al data set.

En quant als models, està clar que el més eficient seria el Random Forest ja que és el que millor pot predir els nous casos de préstec, altres models com el Desicion Tree podrien ser vàlids també però crec que el millor es sense dubtes el Random Forest. Una xarxa neuronal no seria molt adequada en aquest cas ja que tenim una gran quantitat de dades i el temps d'entrenament d'aquesta és molt llarg ja que tarda aproximadament un 60 segons per època, a més, un cop entrenada no es molt eficient a l'hora de predir nous resultats i per tant no seria un bon model de predicció.

En conclusió, per tal de complir amb l'objectiu fixat en aquest treball, decidir si donar o no un préstec a petites empreses, la millor opció és, amb diferència, el model de Random Forest, no només per la seva alta predictibilitat sinó també per el seu temps d'execució i entrenament reduït.