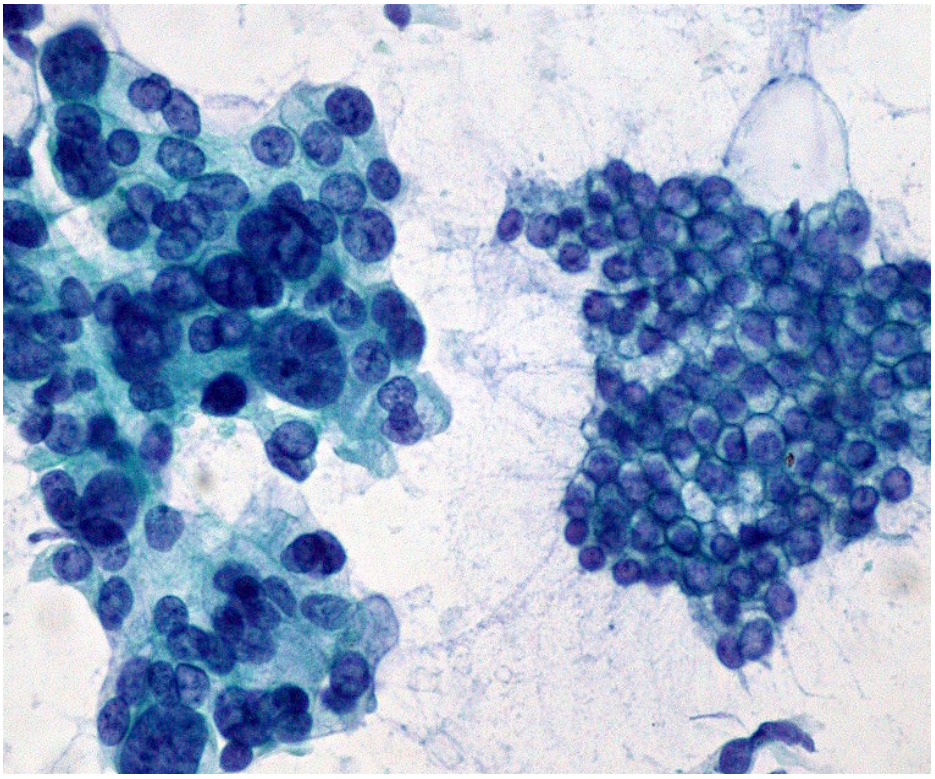
Machine learning for the automated diagnosis of breast cancer



Joan Gomà i Pau Mateo

Grau en Ciència i Enginyeria de Dades

Aprenentatge Automàtic 1

Facultat d'Informàtica de Barcelona, UPC

1 de juny de 2024

ÍNDEX

Introducció **3**
Preprocessament de les dades **5**
Codificació de variables 5
Partició train-test-validation 6
Tractament missing values dades train 6
Tractament missing values dades test 8
Outliers 8
Feature selection **10**
Modelació **12**
Resultats finals i conclusions **16**
Limitacions i línies de futur **17**
Referències **18**

Introducció

El càncer de mama és el tipus de càncer més comú entre les dones de tot el mon. Representa un 25% de tots els casos de càncer i només al 2015 va afectar al voltant de 2.1 milions de persones.

Aquest tipus de càncer es produeix quan les cèl·lules dels teixits mamaris comencen créixer descontroladament (es divideixen més enllà dels límits normals), poden envair altres teixits (intrusió i destrucció dels teixits adjacents) i, a vegades, tenen capacitat de metàstasi (s'estenen a altres punts del cos a través de la limfa o la sang). A la gran major part de càncers i com és al cas de càncer de mama, les cèl·lules cancerígenes formen un tumor maligne (neoplàsia maligna).

Moltes vegades el càncer és detectat, a més de pels símptomes, pel tumor que formen les cèl·lules cancerígenes. Molt sovint els tumors no són cancerígens (tumor benignes), i per això abans de fer cap tractament, cal identificar si un tumor és benigne o maligne. Habitualment es feia duent a terme una biòpsia del tumor per tal de fer un estudi histològic amb el seu grau de diferenciació i d'invasió, i per un estudi molecular per a determinar els seus marcadors biològics i genètics. Però una biòpsia és un procediment quirúrgic molt invasiu que seria preferible evitar fer a menys que sigui com a últim recurs. Una alternativa és utilitzar una tècnica molt menys invasiva, *Fine Needle Aspirations* (FNAs), per extreure una petita quantitat de teixit del tumor.

Mitjançant l'examinació detallada de les característiques de les cèl·lules individuals i aspectes contextuals importants (com el tamany dels agrupaments de cèl·lules), els metges en algunes institucions especialitzades han pogut diagnosticar amb èxit utilitzant FNAs. No obstant això, es considera que moltes característiques diferents estan correlacionades amb la malignitat, i el procés segueix sent altament subjectiu, depenent de la habilitat i experiència del metge. Per tal d'augmentar la velocitat, correcció i objectivitat del procés de diagnòstic, es poden utilitzar tècniques de processat d'imatge i *machine learning* per predir si un tumor és benigne o maligne.

En aquest projecte treballarem amb un dataset proporcionat per la Universitat de Wisconsin que conté 569 mostres de cèl·lules de tumors de les quals s'han extret característiques (individuals i grupals) amb el procediment esmentat, l'objectiu del qual és predir si són cèl·lules malignes o benignes.

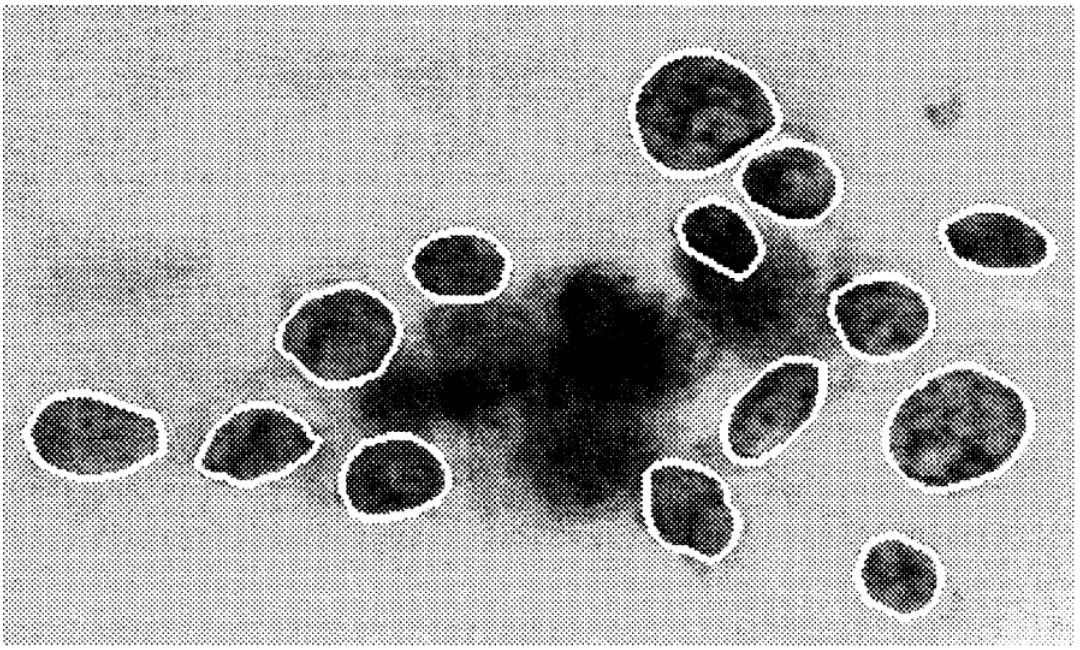


Figura 1: FNA digitalitzada

De cada cèl·lula s'extreuen 10 característiques diferents. Totes aquestes característiques estan modelades tal que numéricament, un valor més gran de la característica indiqui normalment una probabiliat més alta de malignitat del tumor. Les característiques son les seguents:

- ***Radius***
- ***Perimeter***
- ***Area***
- ***Compactness:*** ofereix una mesura de com de compactada està la cèl·lula amb la fórmula $perímetre^2 / àrea.$ D'aquesta manera, aquest valor és minimitzat amb un cercle i augmenta amb qualsevol irregularitat al perímetre. Tot i així, aquesta variable també augmenta per a cèl·lules allargades, cosa que no te perquè indicar més probabilitat a que el tumor sigui maligne. La característica també està biaixada cap amunt per a cèl·lules petites a causa de la disminució de la precisió imposada per la digitalització de la mostra.
- ***Smoothness:*** suavitat del contorn és calculat mesurant la diferencia entre la longitud de les línies radials amb aquesta mateixa longitud mitjana de les línies radials veïnes.

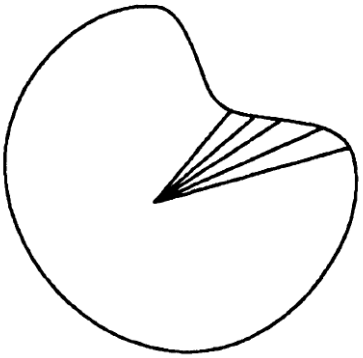


Figura 2: Línies radials usades pel càlcul de la suavitat

- ***Concavity*:** mesura del nombre i la magnitud de les concavitats de la cèl·lula.

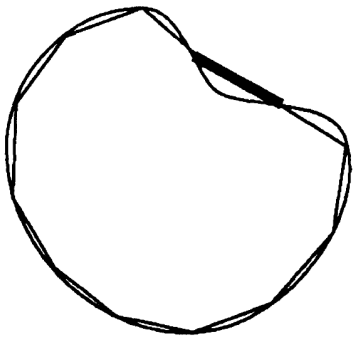


Figura 3: Cordes usades pel càlcul de la concavitat

- ***Concave Points:*** similar a l'anterior variable, però només calcula el nombre de concavitats que té una cèl·lula, en comptes d'atribuïr-li una magnitud.
- ***Fractal Dimension:*** la dimensió fractal d'una figura pot quantificar la seva "complexitat estructural". D'aquesta manera, aproximar la dimensió fractal d'una cèl·lula ens pot ajudar ja que les cèl·lules cancerígenes tendeixen a tenir una major irregularitat i complexitat de forma en comparació amb les cèl·lules normals. En aquest cas, aquesta aproximació es fa usant la *"coastline approximation"* descrita per Mandelbrot [2].
- ***Texture:*** és mesurada amb la variància dels valors en escala de grisos dels píxels que componen l'imatge de la cèl·lula

D'una mateixa mostra d'un tumor obtenim varies cèl·lules, de les quals es computen totes aquestes característiques. A partir d'aquestes s'extreuen 3 variables explicatives de cada característica: la mitjana del valor mesurat a totes les cèl·lules, la variància, i el *worst case,* és a dir, el valor màxim. Això fan un total de 30 variables explicatives de cada observació.

Intuitïvament, les variables més importants seran les *"worst"*, ja que en una mateixa mostra podem trobar que només algunes cèl·lules són malignes, i per tant els valors de les variables *mean* i *se* es veuran afectats.

Preprocessament de les dades

En aquest apartat es mostrarà tot el procés previ que hem fet al dataset per tal de poder utilitzar-lo en models predictius. S'explicaran les transformacions que hem realitzat a les variables perquè siguin operables, les particions que hem efectuat al dataset per tal de poder fer una validació dels nostres models i com s'han tractat els missing values i els outliers.

Codificació de variables

Per tal de fer-nos una idea ràpida de quina forma té el nostra dataset i quines variables s'han de categoritzar o eliminar perquè no serveixen de res, hem utilitzat comandes bàsiques del pandas

per tal de veure totes les variables que teníem i una descripció rapida d'aquestes. Tot i que la sortida ens ha mostrat informació de totes les columnes, les úniques que em considerat que era necessari fer-hi alguna cosa han sigut les següents. Podeu trobar la informació de totes les columnes a l'Annex.

```
#   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Unnamed: 0              569 non-null    int64
 1   id                      569 non-null    int64
 2   diagnosis               569 non-null    object
```

Tant la variable *Unnamed:0* i *id* hem cregut que no estaven aportant cap informació rellevant, ja que es tracten de l'índex i d'un id del qual no tenim cap altra referència. És per això que hem cregut convenient eliminar-les del dataset.
L'altra operació que hem fet ha sigut mapejar la variable *diagonsis*, ja que es tracta del nostre target i ara mateix està prenent valors en "M" o "B". Per tant, l'hem mapejat de tal manera que les observacions malignes se'ls hi correspongui un 1 i a les benignes un 0.

Partició train-test-validation

Per tal de fer un estudi el màxim rigorós possible, s'ha separat el dataset en dues parts: train i test. Al tractar-se de dades mèdiques, hem cregut convenient fer una separació 70%-30%, per així tenir un bon grup de dades de validació i assegurar que els nostres models extrapolaven bé.

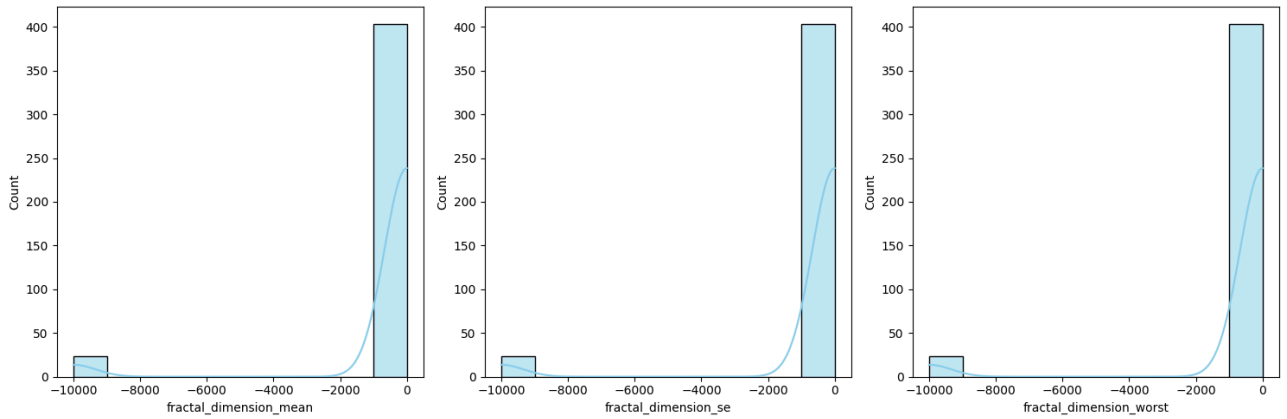
Aquesta operació és primordial per tal d'evitar l'overfitting i tenir models dolents. Separant les dades obtenim la capacitat de poder provar el nostre model amb dades "noves" i, per tant, evaluar el seu rendiment. En el cas que s'obtinguin resultats dolents, s'han d'intentar millorar els models i els seus paràmetres. És molt important fer èmfasi en que en tot el procés de preprocessament i imputacio de missing values no es mirarà la partició test, ja que no volem contaminar aquestes dades amb les de train. En el cas que s'hagin d'imputar valors en la partició de test, ho farem només mirant les dades de tests.

Tractament missing values dades train

En aquesta secció s'explicarà com hem dut a terme la detecció de missing values, com s'han codificat i com s'han imputat.

Detecció de missing values

Tot i que quan hem fet una descripció bàsica del dataset no ens ha sortit cap valor com a null, hem vist que el resum de distribució de quantils, valors màxims i mínims són una mica estranys per les variables *fractal_dimension_mean, fractal_dimension_se, fractal_dimension_worst*. Això és degut a que els missing values estan codificats com a -9999 i no com a *NaN ("Not a Number")*. Per tant, li haurem d'indicar al programa que aquests valors no els tenim per tal que el software els pugui gestionar. Podem confirmar amb els histogrames que aquesta variable té missing values mal imputats.

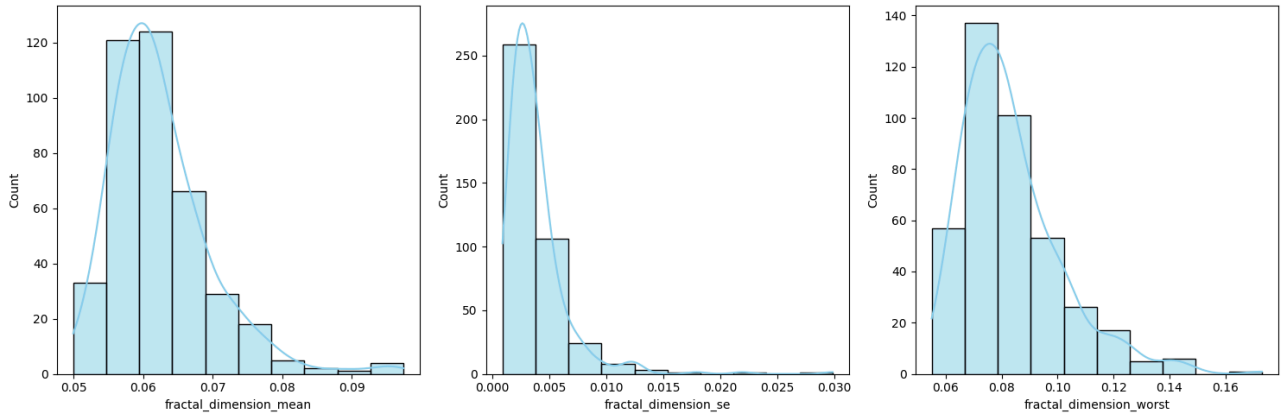


Codificació dels missing values

Per tal de tenir histogrames interpretables sense tenir en compte els missing values s'han hagut de codificar com a tipus *NaN*. Ara mateix la situació en el data set és que aquestes variables tenen menys observacions que les altres, en concret 23 menys.

```
9   fractal_dimension_mean    403 non-null    float64
19  fractal_dimension_se      403 non-null    float64
29  fractal_dimension_worst   403 non-null    float64
```

Tot i això, ara si mostrem els histogrames de les variables, ja son interpretables.



Imputació dels missing values

Ara que ja tenim els missing values identificats correctament el que es farà és imputar els valors utilitzant l'algorisme k-nearest neighbors (k-NN). El que fa aquest algorisme és: donat un hiperparàmetre k que indica el nombre de veïns, troba segons la distància indicada les k observacions del dataset que s'hi assemblen més. Una vegada seleccionats els k veïns més propers, el missing value es reemplaça per algun tipus de resum d'aquests veïns. Si la variable és numèrica, sovint s'utilitza la mitjana o la mediana dels valors dels k veïns. Si la variable és categòrica, es pot utilitzar la moda (el valor més comú) dels k veïns.

En el nostre cas hem fixat l'hiperparàmetre k a 1 i com que la variable que estem imputant és numèrica, agafarem el valor del veí més proper.

Després d'executar el codi podem veure que la imputació dels missing values s'ha efectuat correctament.

```
9   fractal_dimension_mean   426 non-null    float64
19  fractal_dimension_se     426 non-null    float64
29  fractal_dimension_worst  426 non-null    float64
```

Com que els missing values estaven distribuïts sense cap patró pel dataset, no s'observa cap canvi perceptiu en els histogrames de les variables.

Tractament missing values dades test

El procés que se segueix en aquest apartat és idèntic a l'anterior però sense visualitzar les dades en cap moment. Resulta que en el test també tenim missing values en les variables *fractal_dimension_mean, fractal_dimension_se, fractal_dimension_worst.* És per això que primer les codifiquem com a valor nan i després s'efectua la imputació. Dels 143 valors del set de test, només tenim 4 missing values.

```
9   fractal_dimension_mean   139 non-null    float64
19  fractal_dimension_se     139 non-null    float64
29  fractal_dimension_worst  139 non-null    float64
```

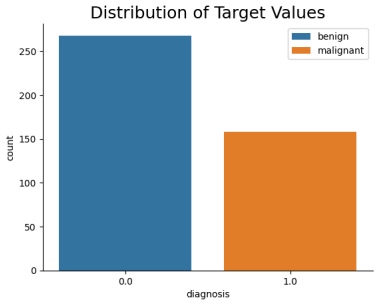
Després d'efectuar la imputació amb l'algorisme K-nn hem obtingut el resultat esperat.

```
9   fractal_dimension_mean   143 non-null    float64
19  fractal_dimension_se     143 non-null    float64
29  fractal_dimension_worst  143 non-null    float64
```

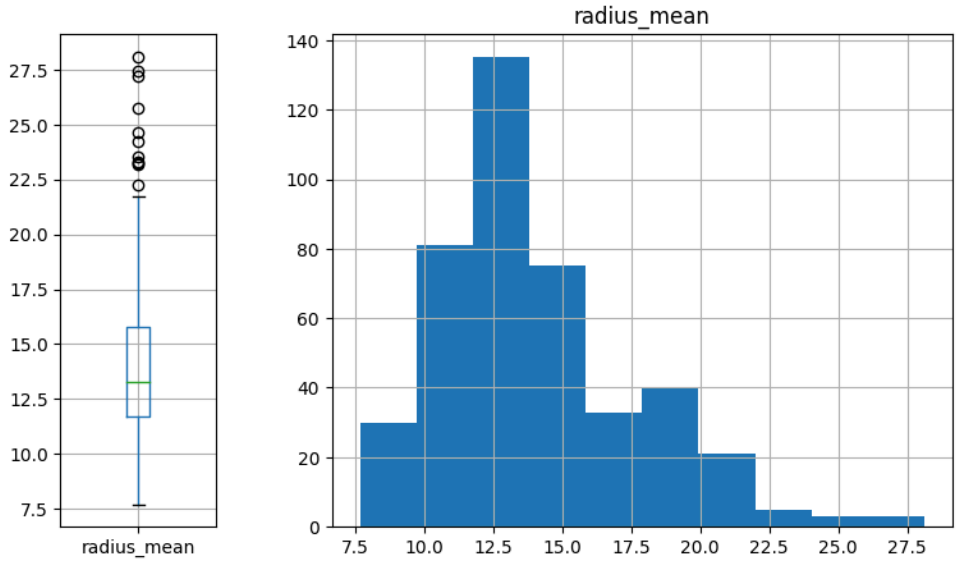
Una vegada ja hem solucionat el tema dels missing values, tractarem a continuació la detecció i tractament dels outliers.

Outliers

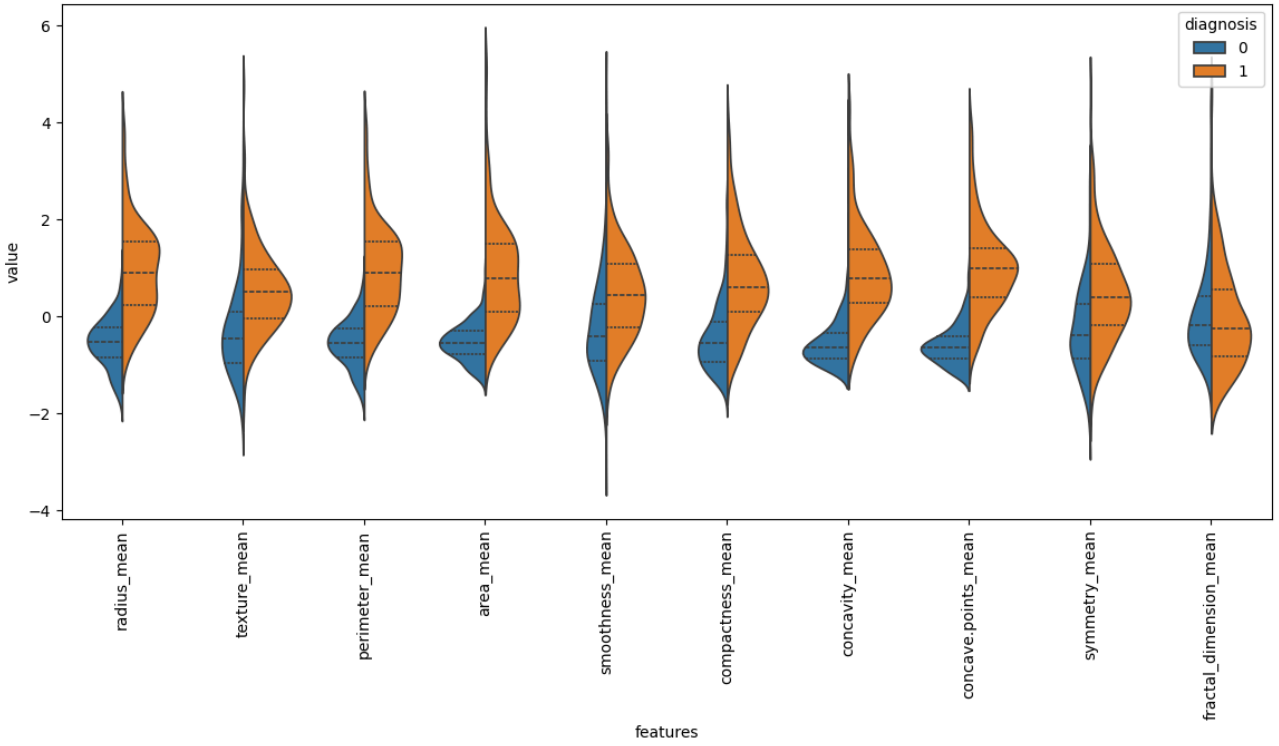
En aquest apartat es veurà la detecció i tractament d'ouliters. Cal recordar que estem treballant amb un dataset sobre cèl·lules cancerígenes i que, per tant, les observacions de casos malignes, no es poden ignorar per molt que estiguin fora dels estàndards. En el cas que ho féssim, estaríem empitjorant els nostres models, ja que no tindrien tota la informació. És per això que haurem d'anar molt amb compte a l'hora d'eliminar dades. A part, tal i com s'ha comentat a la introducció i com es pot visualitzar a la següent gràfica, la presència d'observacions on el target és maligne és inferior al de benignes. És per això que si treguessim moltes observacions de casos malignes ens quedaria un dataset massa desequilibrat.



Vegem per exemple l'anàlisi de la variable *radius_mean* per fer-nos una idea i després ja farem un estudi més generalitzat per totes les variables.



Si mirem l'histograma d'aquesta variable i el plot que ens mostra els outliers (grafica de l'esquerra) es pot observar que aquesta variable té presència d'outliers. Ara bé, quan executem el codi per veure quin diagnòstic tenen aquests outliers ens trobem que tots es tracten de cèl·lules cancerígenes. És per això, que no hem tret cap observació a la lleugera per tal de no tenir un impacte no esperat en els nostres models. De manera generalitzada, podem veure els següents *violin plots* que inclouen les pdf de totes les nostres variables contrarestades directament amb la seva diagnosis. Els plots corresponents a la resta de variables són molt similars, i els podeu trobar a l'Annex (Gràfiques 1 i 2).



El que es pot veure de manera generalitzada és que en la majoria de casos, els valors més extrems es corresponen a observacions que estan catalogades com a cèl·lules malignes. És per això que hem decidit no treure les variables marcades com a outlier ja que, com s'ha comentat abans, no ens interessa treure observacions de cèl·lules malignes per tal de no desaquilibrar encara més el dataset.
Apart d'això, també podem veure en aquestes gràfiques quines variables són més importants per predir la diagnosis. En el cas que les dues corbes d'una variable siguin pràcticament idèntiques, molt probablement, no serà gaire útil per tal de predir la diagnosis.

Feature selection

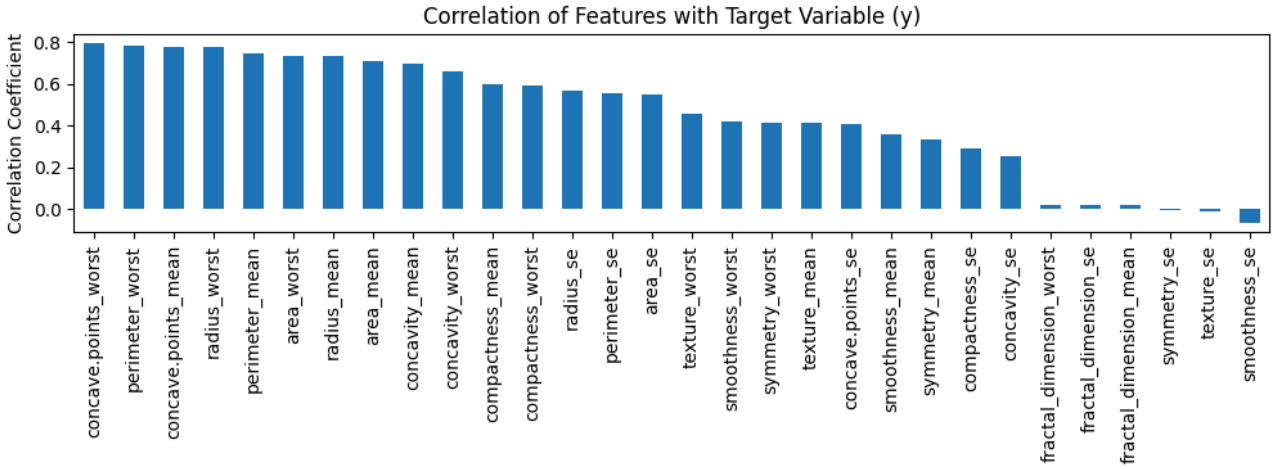
Com que el dataset amb el que estem treballant té moltes variables (en concret 30) i 569 observacions, hem decidit que calia fer una abstracció del dataset per tal de trobar quines variables eren les més útils per tal de no complicar el modelatge.

Al tractar-se de dades de mesures sobre conceptes similars en molts casos, creiem que hi haurà molts conjunts de variables amb alta correlació, sobretot entre els triplets de variables *mean*, *se* i *worst*. Una forma fàcil de veure les correlacions entre les variables és mirant la matriu de correlacions. Vegeu la Gràfica 3 (*Correlation Map*) per tal d'entendre l'anàlisi que fem a continuació.

Tal i com ens esperàvem, moltes variables estan altament correlades entre elles. De fet, es poden identificar fins i tot les diagonals (a part, evidentment, de la diagonal principal) que corresponen a les correlacions entre els triplets de variables mencionades, que tendeixen a tenir una alta correlació.

Un altre fet interessant és que no s'observa que hi hagi cap variable que estigui negativament correlada a la variable diagnosis (excepte alguns casos, però són correlacions molt baixes). Tal i com s'havia comentat a la part d'introducció al dataset que les variables estaven dissenyades tal que més valor de la variable impliqués més probabilitat a ser maligne, i així és compleix.

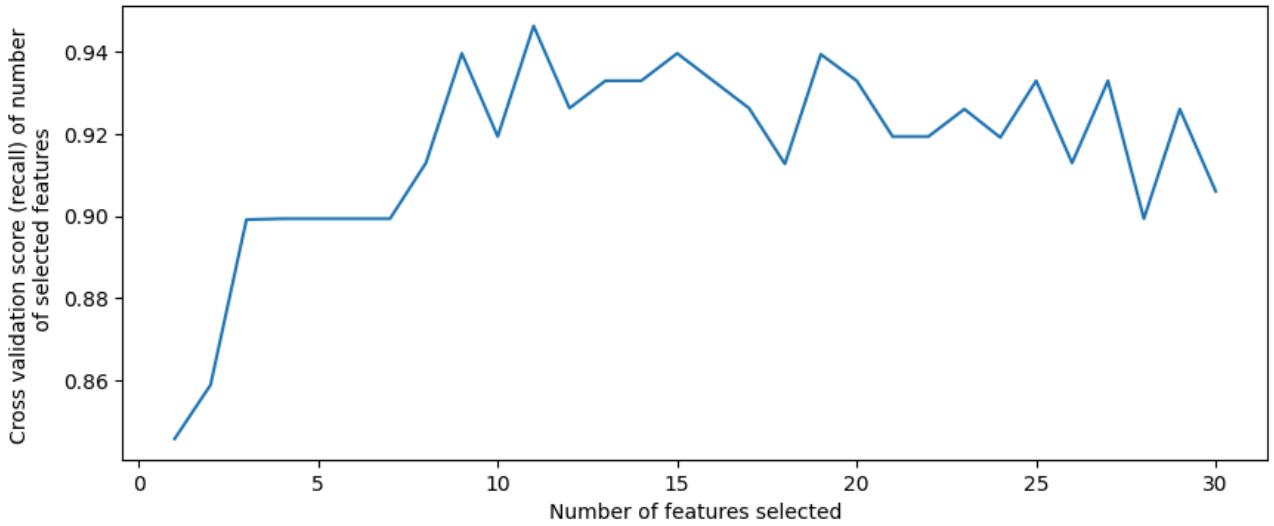
Abans de fer *Feature Selection* hem decidit fer una selecció prèvia: les variables molt poc correlacionades amb la variable *target* ja no les considerarem a partir d'aquest punt. La següent gràfica mostra les correlacions entre totes les variables amb la variable *target*, en ordre descendent.



Hem descartat doncs les variables que tenen molt poca correlació amb el *target*. Això són: *fractal_dimension_mean, texture_se, smoothness_se, fractal_dimension_se, symmetry_se* i *fractal_dimension_worst.* Amb això continuem a aplicar Feature Selection.

Una manera amb la que podríem haver fet *Feature Selection* és eliminar variables que estiguin altament correlades entre elles, ja que en el nostre cas, com hem vist, en podríem descartar bastantes. Per exemple, podríem descartar dues de les variables *radius_mean*, *perimeter_mean*, *area_mean* (quedant-nos amb *perimeter_mean*, ja que és la que més correlació té amb la variable target), ja que aquestes estan altament correlades entre elles. I així podríem treure també bastantes altres variables: *perimeter_worst, compactness_worst, concave.points_worst, perimeter_se, concavity_se*... Amb només això ja fem una bona i simple reducció de la dimensionalitat de les dades. Vegeu Gràfica 4 a l'Annex per veure el nou *Correlation Map*.

No obstant la simplicitat d'aquest mètode, nosaltres ens hem decantat per una opció més robusta. Hem utilitzat el mètode de *Recursive Feature Elimination with Cross Validation* (RFECV). Aquest mètode utilitza un model d'aprenentatge supervisat relativament senzill (nosaltres usarem *Random Forest*) i validació creuada per determinal el millor nombre de variables i quines a seleccionar. Recursivament segueix el següent procés: entrena el model amb totes les característiques que li queden, utilitza *Cross Validation* per determinar l'*accuracy* (o qualsevol altra mètrica) obtingut i ordena les variables de més rellevants a menys. Després, elimina la variable menys rellevant i repeteix el procés fins que es queda sense variables. El que fa el programa és guardar-se el valor del models per cada pas, de tal manera que quan acaba podem veure amb quin nombre de paràmetres s'obté millor resultat. Hem decidit fer-ho amb la mètrica *recall*, ja que, com a la major part de datasets clínics, és molt més costós un fals negatiu que un fals positiu. La següent gràfica mostra els resultats obtinguts amb *RFECV*:




```
Optimal number of features : 11
'texture_mean',   'perimeter_mean', 'area_mean',      'concave.points_mean',
'area_se',        'radius_worst',   'perimeter_worst','area_worst',
'concavity_worst','concavity_mean', 'concave.points_worst'
```

Podem veure dos pics en l'índex 9 i 11. Com que l'11 és lleugerament superior, la funció per defecte ens recomana quedar-nos amb les 11 variables que veiem a l'output. Nosaltres seguirem l'estudi amb aquestes 11, però amb 9 segurament s'obtindrien resultats igual de bons. En tal cas, les 9 variables serien:

```
'perimeter_mean', 'concavity_mean',  'concave.points_mean',
'radius_worst',   'texture_worst',   'perimeter_worst',
'area_worst',     'concavity_worst', 'concave.points_worst'
```

Modelació

Per a aquest projecte hem considerat i ajustat els següents models: LDA, QDA, k-NN, Naive Bayes, Decision Tree, Random Forest, Logistic Regression i dos models de Boosting: AdaBoost i XGBoost.

Per estimar l'error de generalització dels models, usarem *Cross Validation* sobre el conjunt de train amb *folds* de tamany 5, i amb aquests resultats triarem el millor (o millors) models, els quals els avalarem de nou amb el set de test. Tot i que hem fet *Feature Selection* per reduïr la dimensionalitat de les dades, no només entrenarem els models amb les variables seleccionades: també els entrenarem amb totes les variables per tal de veure com afecta als diferents models el fet de tenir variables molt correlades i l'impacte del *Feature Selection*.

Haureu notat que a la part de preprocessament no hem aplicat cap normalització a les dades. I certament, com que diverses variables tenen rangs de valor bastant diferents, per a alguns models és important que les dades estiguin normalitzades. Per tant, també ajustarem els models amb les dades estandaritzades[1]. Dels models que considerem, els que més sensibles a la estandardització són k-NN i Logistic Regression.

Els models els hem ajustat amb els següents paràmetres (molts d'ells són per defecte):

- **LDA**: solver 'svd'
- **QDA**: paràmetre de regularització nul
- **k-NN**: pesos uniformes, *leaf size* de 30, mètrica de Minkowski amd p=2, i k=5, basant-nos ens els resultats obtinguts de validació creuada (*10-fold*) per a diferents valors de k. Vegeu les gràfiques 5 i 6 de l'Annex.
- **Naive Bayes**: *variance smoother* de 1e-9
- **Decision Tree**: criteri *gini*
- **Random Forest**: criteri *gini*, 100 estimadors
- **Logistic Regression**: *penalty* de L2
- **AdaBoost**: màxims estimadors de 50, *learning rate* de 1.
- **XGBoost**: 100 estimadors, *learning rate* de 0.3, booster 'gbtree'.

A continuació mostrem els resultats de validació obtinguts per a les tres dades diferents: *all features*, *selected features* i *standardized selected features*. Hem avaluat els models amb vàries mètriques, i els hem ordenat descendentment en funció del *recall*.

Durant tot aquest anàlisi i comparació de models, farem servir sobretot el *recall*, ja que amb aquest dataset és molt més 'car' un fals negatiu que un fals positiu: en cas d'equivocar-se al predir si un pacient té càncer o no, més val equivocar-se i dir que sí que en té quan realment no, que dir que no en té quan realment sí que en te.

[1] Només ho farem amb les variables obtingudes al *Feature Selection*; el dataset amb totes les variables no provarem a estandaritzar-lo.

All features:

	accuracy	balanced accuracy	f1 score	precision	recall	roc auc
QDA	0.954652	0.954118	0.940541	0.930031	0.952644	0.990925
XGBoost	0.964842	0.959729	0.952518	0.966207	0.939540	0.988961
Random Forest	0.949747	0.945022	0.932160	0.939230	0.926207	0.985745
AdaBoost	0.954778	0.947688	0.938842	0.959310	0.919540	0.985980
DecisionTree	0.919620	0.915607	0.893848	0.888986	0.899540	0.915607
LogisticReg	0.942152	0.933492	0.921396	0.945255	0.899310	0.987218
NaiveBayes	0.937184	0.928051	0.913941	0.940893	0.892184	0.984806
LDA	0.949842	0.937063	0.929128	0.978553	0.886207	0.986995
KNN	0.914525	0.897821	0.879008	0.933523	0.831724	0.945795

Selected Features:

	accuracy	balanced accuracy	f1 score	precision	recall	roc auc
AdaBoost	0.959715	0.955648	0.946066	0.953504	0.939540	0.981797
QDA	0.957278	0.953574	0.942752	0.947527	0.939310	0.990951
DecisionTree	0.952278	0.948314	0.936731	0.943069	0.932874	0.948314
XGBoost	0.954778	0.950240	0.939364	0.947312	0.932644	0.977066
Random Forest	0.959810	0.951614	0.944906	0.972414	0.919310	0.987977
LDA	0.957310	0.944207	0.939692	0.992593	0.892414	0.988728
NaiveBayes	0.932120	0.917155	0.903407	0.958004	0.858391	0.989215
LogisticReg	0.914494	0.901699	0.881673	0.918973	0.851724	0.982089
KNN	0.914525	0.897821	0.879008	0.933523	0.831724	0.945795

Standardized selected features:

	accuracy	balanced accuracy	f1 score	precision	recall	roc auc
LogisticReg	0.964810	0.960981	0.952827	0.960774	0.946207	0.990580
AdaBoost	0.959715	0.955648	0.946066	0.953504	0.939540	0.981797
KNN	0.964842	0.959655	0.952240	0.967270	0.939310	0.976934
QDA	0.957278	0.953574	0.942752	0.947527	0.939310	0.990951
XGBoost	0.954778	0.950240	0.939364	0.947312	0.932644	0.977066
Random Forest	0.962310	0.955063	0.948665	0.972627	0.926207	0.987253
DecisionTree	0.939715	0.932866	0.918731	0.934445	0.905977	0.932866
NaiveBayes	0.944715	0.936718	0.924088	0.945272	0.905517	0.986585
LDA	0.957310	0.944207	0.939692	0.992593	0.892414	0.988728

Com es pot veure, en general tots els models obtenen bons resultats: excepte els models KNN i Decision Tree, tota la resta obtenen, per als tres casos, un valor de AUC superior a 0,97.

Comencem comparant, en general, la diferència entre usanr totes les variables, usar les variables seleccionades amb *Feature Selection* i aquestes mateixes però aplicant estandarització.
Pràcticament no hi ha diferència entre els resultats obtinguts amb totes les variables i amb les variables seleccionades. Més o menys, tots els models obtenen les mateixes mètriques de validació per als dos casos. Pel que fa al *recall*, destaca el cas del QDA, que amb totes les variables obté un *recall* de 0.95, que és el més alt amb diferència, però amb les variables seleccionades, no hi ha cap model que superi el 0,93. Tot i així, sembla que amb les variables seleccionades amb *Feature Selection* hi ha més bons resultats de *balanced accuracy* i *F1 score*.

Des d'un punt de vista computacional, és clar que ajustar els models amb totes les variables és molt més costós que fer-ho amb les variables seleccionades. Per comparar-ho, el temps d'execució de l'entrenament i validació de tots els models amb totes les variables és de 12 segons, i amb només les variables seleccionades de 7,6 segons. Per tant, podem afirmar que el *Feature Selection* que hem aplicat ha donat bons resultats, ja que amb menys variables aconseguim obtenir resultats igual de bons.

Comparant la segona i tercera taula, veiem que els models que es veuen més beneficiats de l'estandarització són KNN i Logistic Regression, tal com era d'esperar. La millora és tan gran que aquests passen d'estar útlim i penúltim (en quant a *recall*) a primer (Logistic Regression) i tercer (KNN). La resta de models obtenen pràcticament els mateixos resultats que sense estandaritzar.

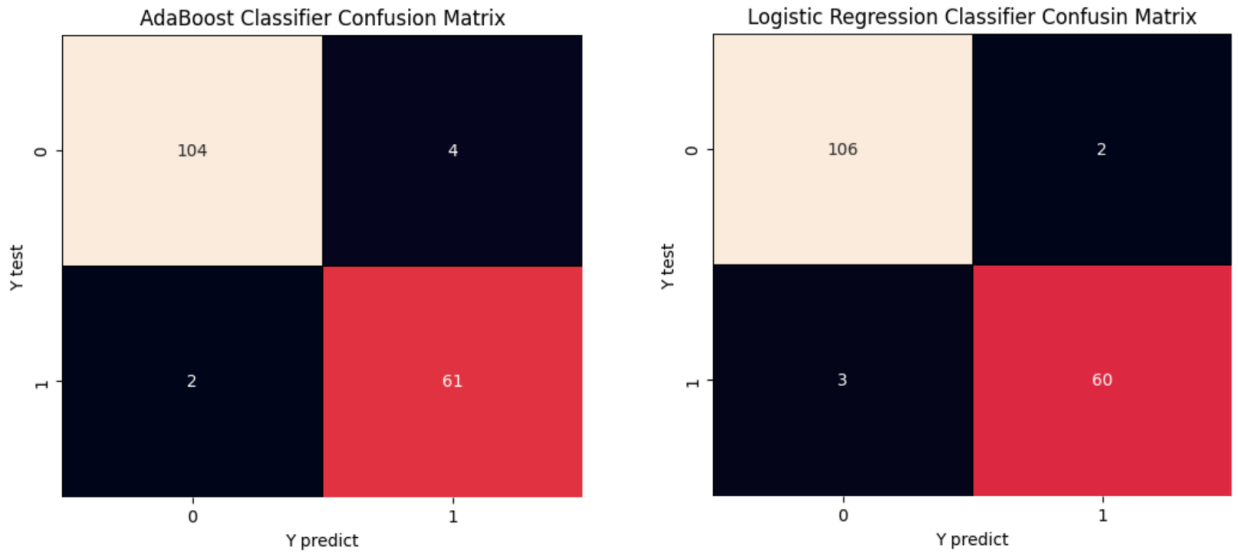
I per últim, cal mencionar que pel que fa a la precisió, el model LDA sobrepassa a tots els altres en tots tres casos. Malhauradament, no obté tants bons resultats de *recall* i de la resta de mètriques.

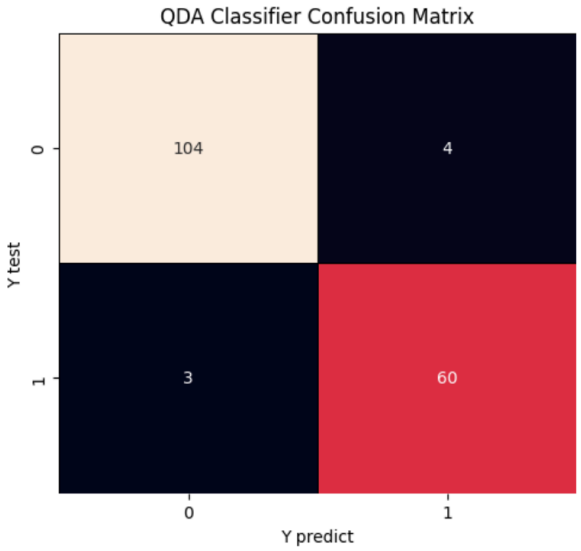
De totes aquestes combinacions de models / dataset, considerem que les millors són:

- QDA amb totes les variables
- AdaBoost amb les variables obtingudes de la *Feature Selection*
- Logistic Regression, amb estandarització i les variables de la *Feature Selection*

A continuació entrenem aquests tres models amb tot el conjunt d'entrenament, i els validem amb el conjunt de test que ens havíem guardat per estimar l'error de generalització. Es poden veure les seves matrius de confusió respectives i el resum de la performance dels tres models.

	accuray	balanced accuracy	f1_score	precision	recall
AdaBoost	0.964912	0.965608	0.953125	0.938462	0.968254
LogisticReg	0.970760	0.966931	0.960000	0.967742	0.952381
QDA	0.959064	0.957672	0.944882	0.937500	0.952381





Resultats finals i conclusions

Els tres models han obtingut resultats molt bons sobre el conjunt de test, fins i tot millors dels que havien obtingut abans amb la validació creuada sobre el conjunt d'entrenament. El model de regressió logística és el que obté millors mètriques excepte per al *recall*. El model AdaBoost obté un 96.82% de *recall*, i QDA i Logistic Regression obtenen 95.23%.

En aquest projecte hem vist que hi ha moltes maneres de tractar les dades i d'enfocar el preprocessat, moltes d'elles igual de vàlides i amb resultats similarment bons, i que és important tenir-ho en compte a l'ajustar els models i saber què convé per a cada un d'ells. Hem après a tractar amb dades reals i a contextualitzar el tractament d'aquestes i la interpretació dels resultats en un àmbit mèdic.

Els models que hem proposat són: *Logistic Regression* (millor rendiment amb estandardització i *Feature Selection*), *AdaBoost* (millor rendiment amb *Feature Selection*) i *QDA* (millor rendiment amb totes les variables), que obtenen tots tres un *accuracy* i un *recall* superior a 95%.

Amb aquests models ens podem considerar capaços de classificar un tumor de mama com a maligne o benigne a partir de característiques obtingudes d'una imatge digital de les cèl·lules d'aquest tumor. El gran avantatge que representa això en l'àmbit mèdic és que, tal com hem explicat a la introducció del projecte, es pot utilitzar la tècnia de *Fine-Needle aspiration* per obtenir cèl·lules del tumor d'un pacient, que és una tècnica molt menys invasiva i ràpida que l'avaluació del creixement cel·lular a partir de l'obtenció de les cèl·lules amb una biòpsia.

Tot i així, cal dir que la única manera d'assegurar que un tumor és benigne o benigne és avaluant el seu creixement cel·lular (és a dir, s'ha de fer una biòpsia). Però això no treu importància a aquesta tècnica, ja que podria ser utilitzada com a diagnòstic previ (és molt més ràpid que avaluar el creixement cel·lular) o com a diagnòstic alternatiu a altres tècniques. I per suposat, en tot moment un professional hauria de interpretar els resultats i decidir quan convé utilitzar aquesta tècnica i quan el diagnòstic és o no fiable.

Limitacions i línies de futur

Tot i que els resultats que hem obtingut son força bons, creiem que en alguns aspectes del treball tenim marge de millora i explorar nous aspectes de cara al futur que segurament no hem tingut tant en compte.

Una d'aquestes millores creiem que es podria aplicar en l'apartat de feature selection ja que, tot i que hem fet un anàlisi bastant extens, hauríem pogut aplicat eines com ara l'Anàlisi de Components Principals (PCA) per tal de fer una millor selecció i extracció de característiques. L'anàlisi del PCA ens permetria reduir la dimensionalitat del conjunt de dades, mantenint la major part de la variabilitat i informació rellevant, la qual cosa podria resultar en models més eficients i amb un millor rendiment.

Un altre aspecte a explorar seria l'entrenament d'altres models com podrien ser xarxes neuronals i *support vector machines*. Tot i que hem fet ús d'un gran nombre de models sempre es pot ampliar ja que, com s'ha vist, cada un té les seves fortaleses i debilitats i com més context general es prengui sobre els resultats, millor.

Al tractar-se d'un dataset de classificació on no tenim la mateixa distribució per les dues variables de target, hauriem pogut aplicar tècniques de *class imbalance* per tal d'aconseguir millors models. En un futur, podríem explorar tècniques com el sobremostreig (oversampling) i el submostreig (undersampling), així com l'ús de mètriques específiques per a dades desequilibrades, per tal de mitigar aquest efecte i obtenir prediccions més precises. Com que el nostre dataset tampoc estava molt desequilibrat hem pogut obtenir bons models igualment, però el tractament d'aquest desequilibri creiem que encara ens milloraria més el rendiment dels models.

Per últim, creiem que es podria fer una exploració més àmplia en quant als hiperparàmetres que fixem en els nostres models. Per exemple, podríem fer un anàlisi més extens en quant als hiperparàmetres del learning rate en l'algoritme AdaBoost, així com altres paràmetres crítics que podrien influir en la seva capacitat d'aprenentatge i generalització.

Referències

[1] W. Nick Street, William H. Wolberg and 0. L. Mangasarian. *Nuclear feature extraction for breast tumor diagnosis.* Departments of Computer Sciences, Surgery, and Human Oncology University of Wisconsin, Madison, WI 53706

[2] B.B. Mandelbrot. *The fractal geometry of nature.* W.H. Freeman and Company, New York, NY, 1977.

[3] S. Tounsi, I. F. Kallel and M. Kallel, *"Breast cancer diagnosis using feature selection techniques"*, 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 2022, pp. 1-5, doi: 10.1109/IRASET52964.2022.9738334.

[4] User Guide. (n.d.). Scikit-learn. https://scikit-learn.org/stable/user_guide.html

[5] Python Package Introduction — *xgboost 2.0.3 documentation*. (n.d.). https://xgboost.readthedocs.io/en/stable/python/python_intro.html

[6] UCI Machine Learning Repository. (n.d.). https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic