APC Practica 2

David Candela (1563873), Alex Casamitjana (1568143), Guillermo Raya (1568864)

13 de desembre de 2021

Índex

1 Apartat B

1.1 Classificació per diversos valors de C

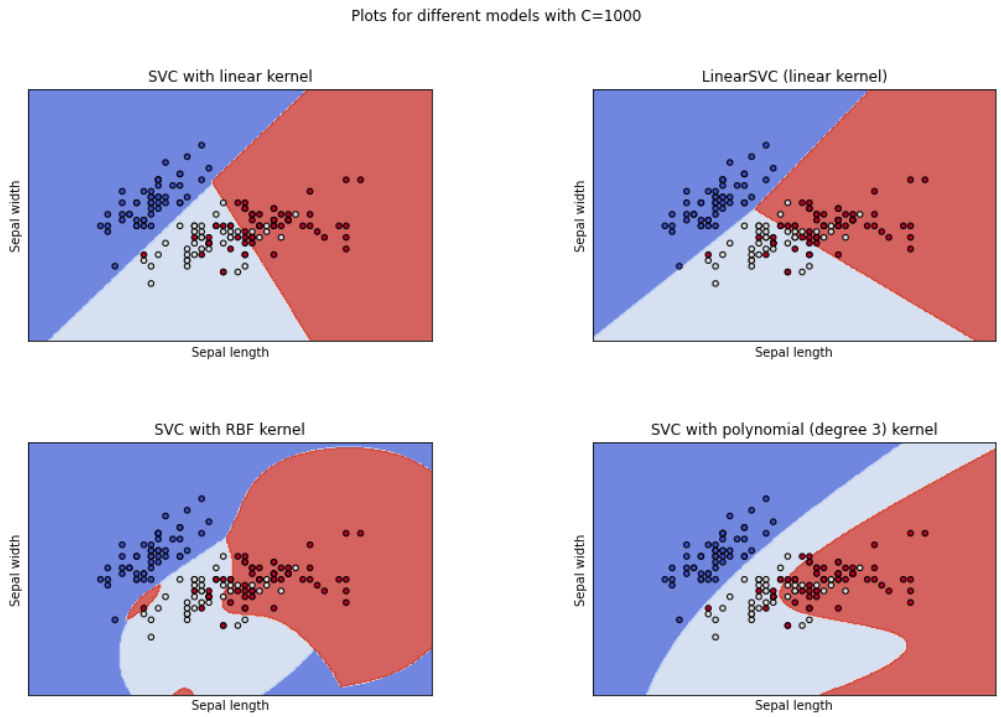


Figura 1: Plot per C = 1000

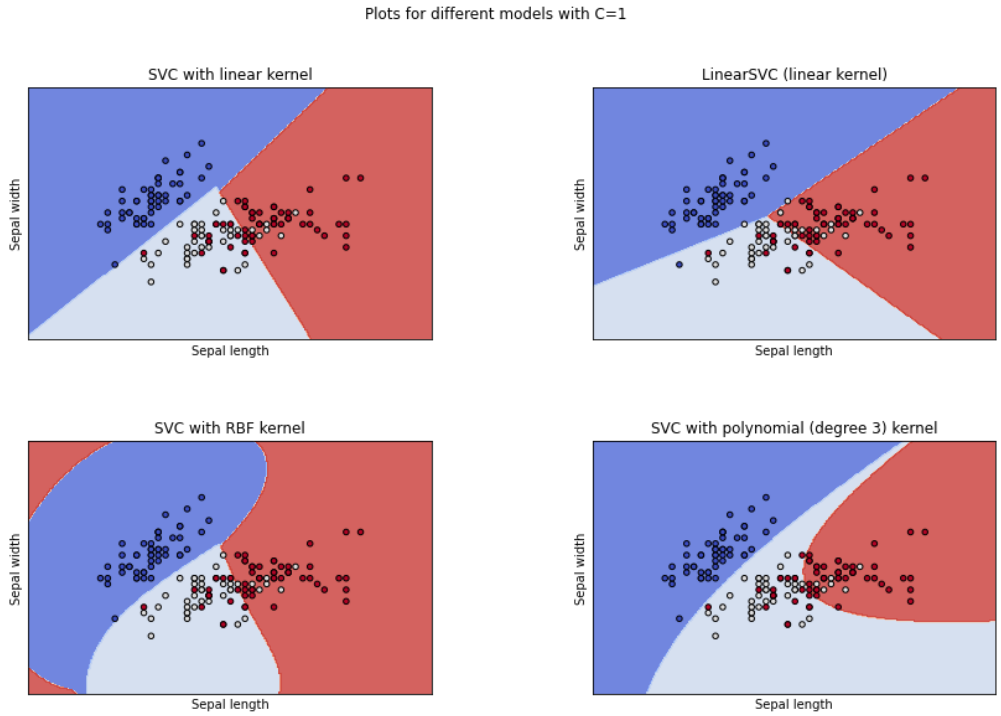


Figura 2: Plot per C = 1

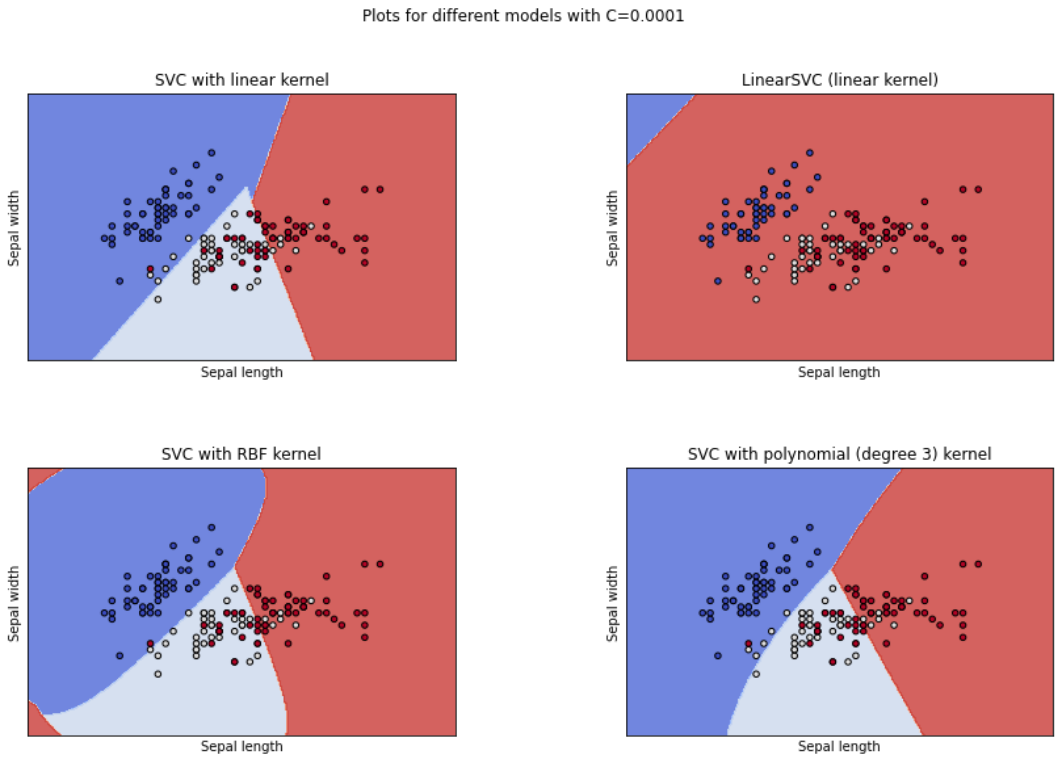


Figura 3: Plot per C = 0.0001

Com es pot apreciar als plots, per Cs petites, els kernels lineals no treballen massa bé mentre que el polinomial i el RBF donen bons resultats. Per valors grans de C els kernels lineals donen bones prediccions mentre que el RBF i el polinomial comencen a fer overfitting.

1.2 Classificació per diversos valors de gamma

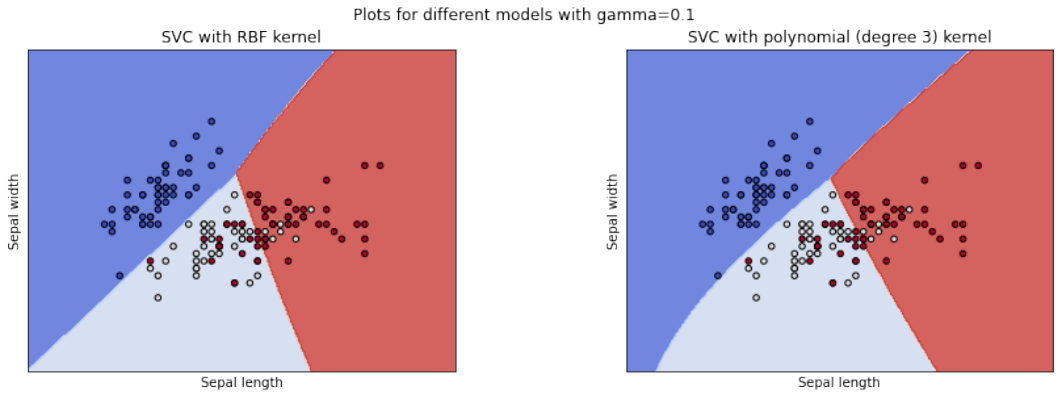


Figura 4: Plot per gamma = 0.1

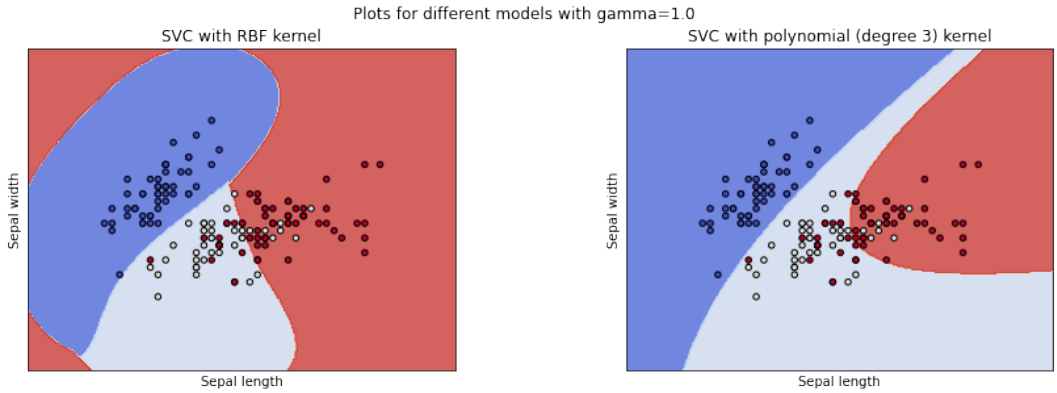


Figura 5: Plot per gamma = 1.0

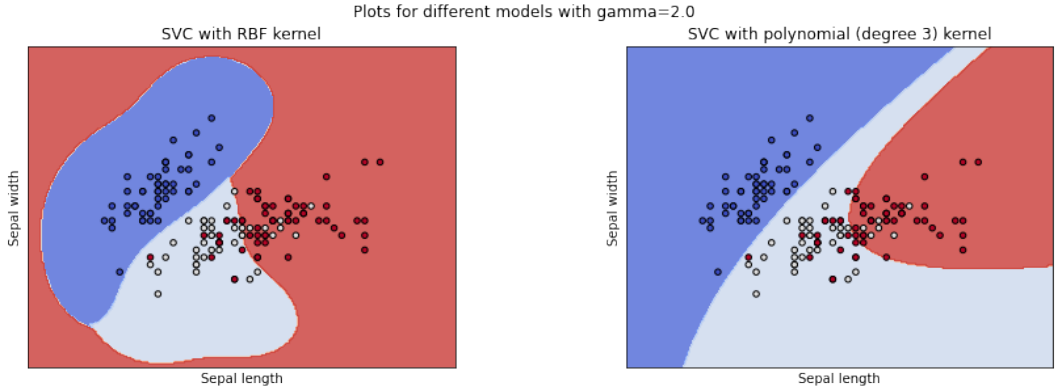


Figura 6: Plot per gamma = 2.0

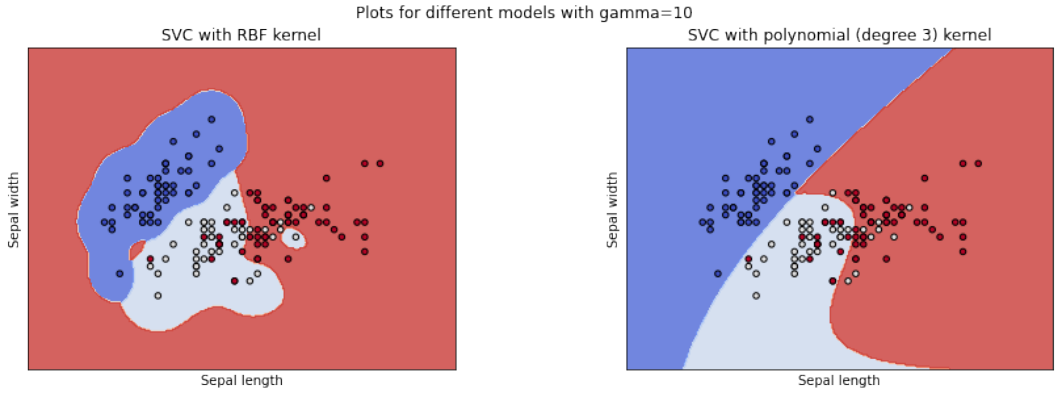


Figura 7: Plot per gamma = 10.0

Als plots anteriors podem observar que

1.3 Classificació per diversos valors de degree

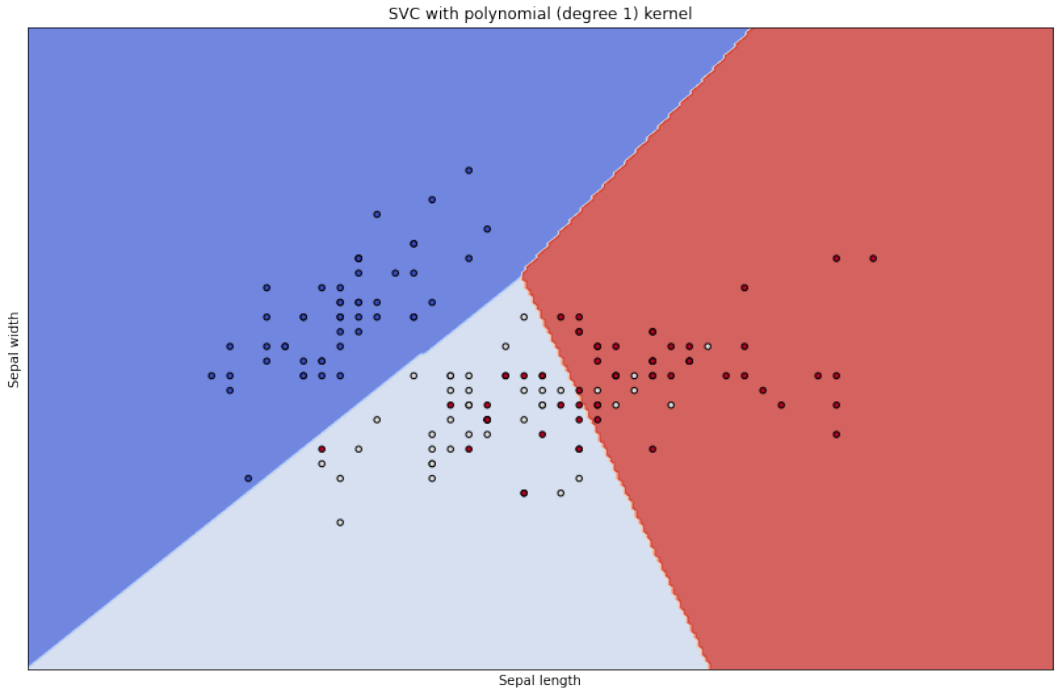


Figura 8: Plot per degree = 1

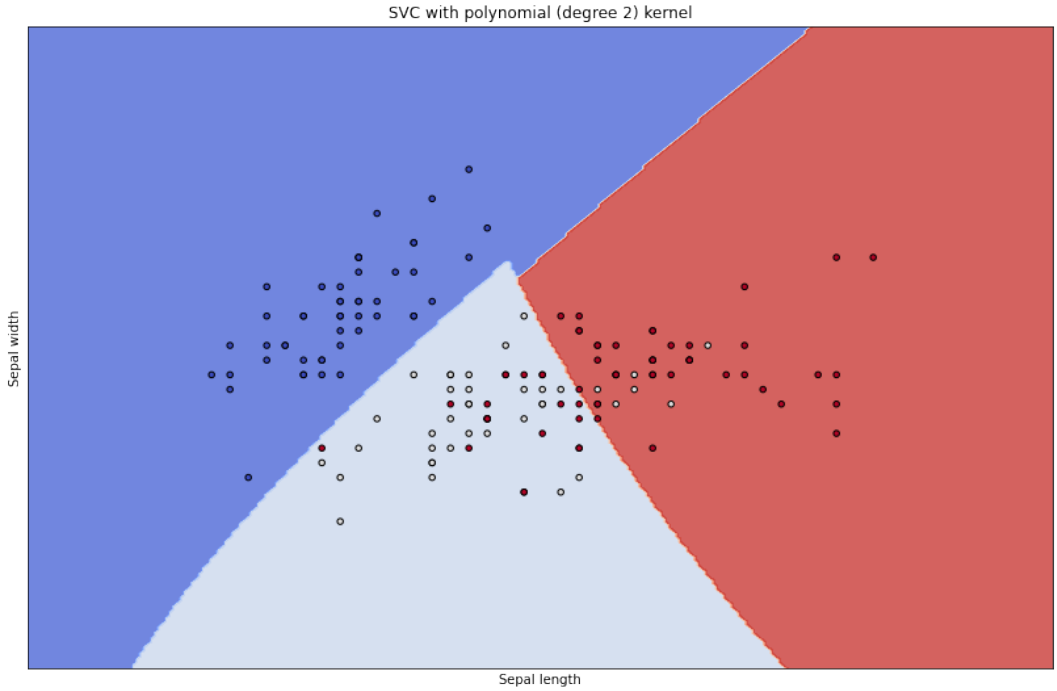


Figura 9: Plot per degree = 2

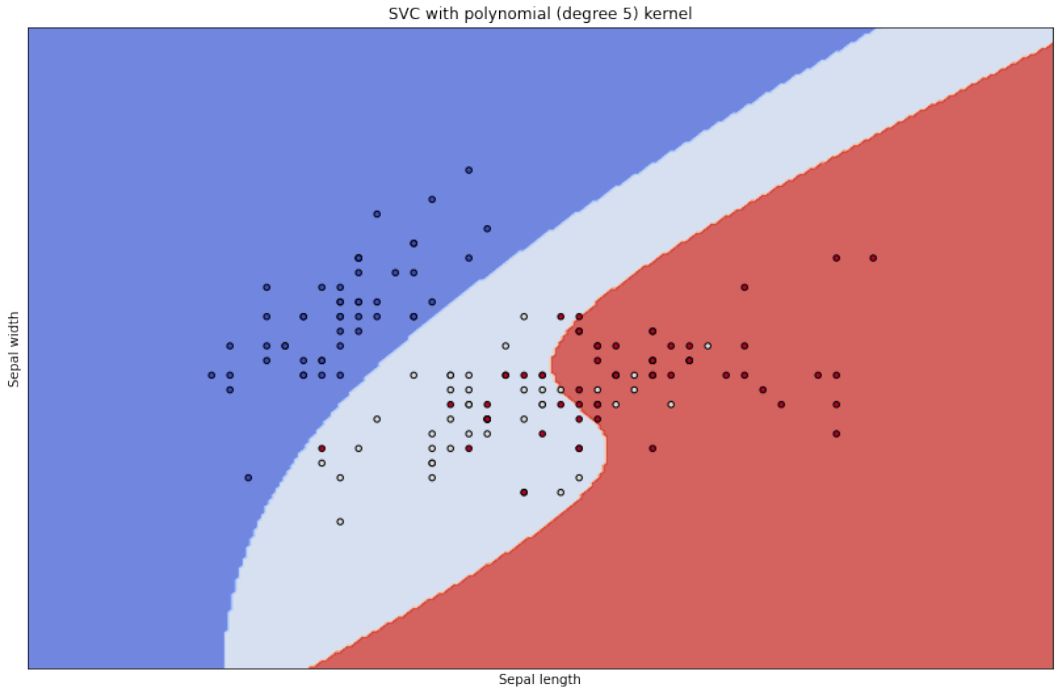


Figura 10: Plot per degree = 5

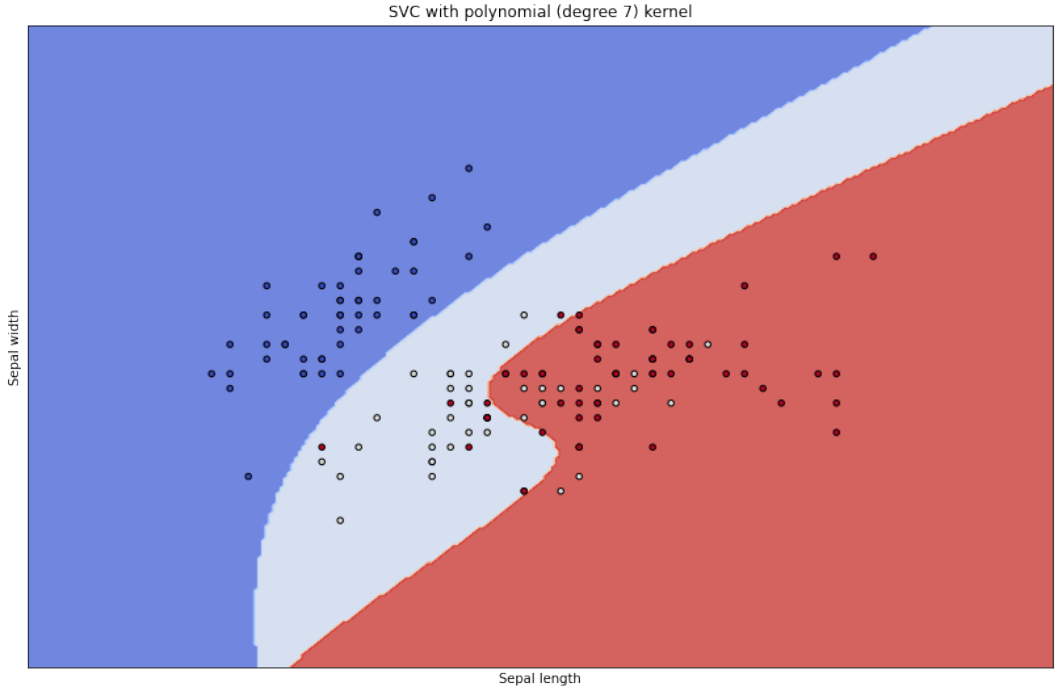


Figura 11: Plot per degree = 7

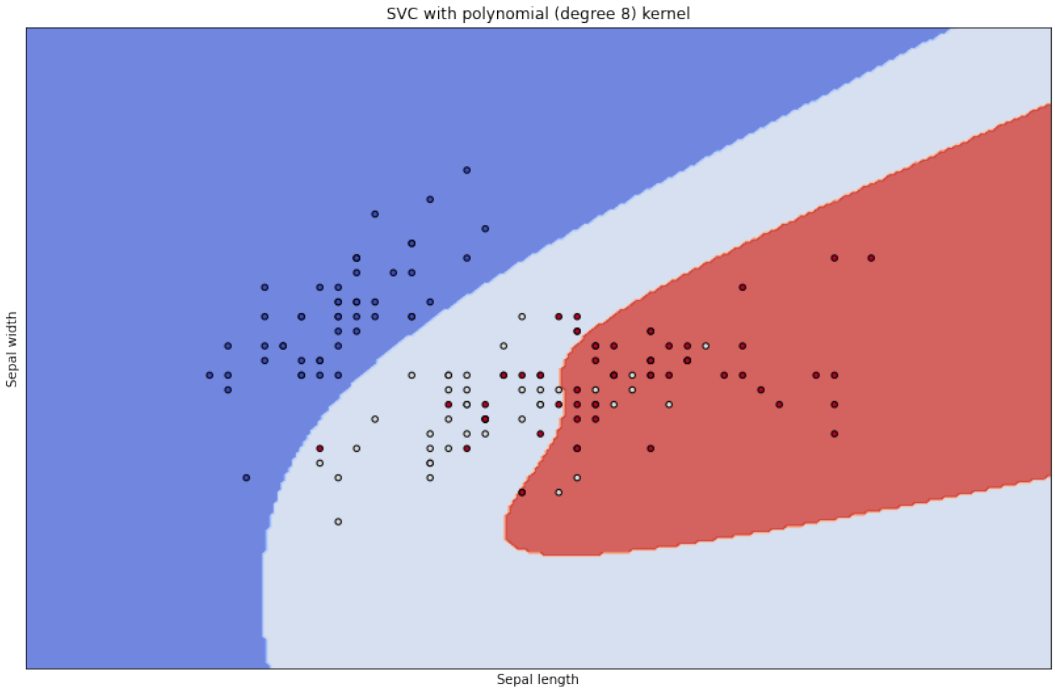


Figura 12: Plot per degree = 8

2 Apartat A

2.1 EDA (exploratory data analysis)

- Quants atributs té la vostra base de dades?
  La nostra base de dades té 41 atributs.

- Quin tipus d'atributs tens? (Numèrics, temporals, categòrics, binaris...)
  Els tipus d'atributs que tenim són els següents:

Atribut	Tipus	Descripció
name	String	The English name of the Pokemon
japanese_name	String	The Original Japanese name of the Pokemon
pokedex_number	Numeric	The entry number of the Pokemon in the National Pokedex
percentage_male	Numeric.	The percentage of the species that are male. Blank if the Pokemon is genderless.
type1	String categoric	The Primary Type of the Pokemon
type2	String categoric	The Secondary Type of the Pokemon
classfication	String categoric	The Classification of the Pokemon as described by the Sun and Moon Pokedex
height_m	Numeric	Height of the Pokemon in metres
weight_kg	Numeric	The Weight of the Pokemon in kilograms
capture_rate	Numeric	Capture Rate of the Pokemon
base_egg_steps	Numeric	The number of steps required to hatch an egg of the Pokemon
abilities	String categoric	A stringified list of abilities that the Pokemon is capable of having
experience_growth	Numeric	The Experience Growth of the Pokemon
base_happiness	Numeric	Base Happiness of the Pokemon
against_?	Numeric	Eighteen features that denote the amount of damage taken against an attack of a particular type
hp	Numeric	The Base HP of the Pokemon
attack	Numeric	The Base Attack of the Pokemon
defense	Numeric	The Base Defense of the Pokemon
sp_attack	Numeric	The Base Special Attack of the Pokemon
sp_defense	Numeric	The Base Special Defense of the Pokemon
speed	Numeric	The Base Speed of the Pokemon
generation	Numeric	The numbered generation which the Pokemon was first introduced
is_legendary	Binary	Denotes if the Pokemon is legendary

- Com es el target, quantes categories diferents existeixen?
  El nostre target és 'is_legendary', i, com que és una dada binària, té dues categories (una per a Pokémon llegendaris, l'altra per a no llegendaris).

- Podeu veure alguna correlació entre X i y?
  Si. A continuació el gràfic de correlació i els histogrames de les variables més importants:

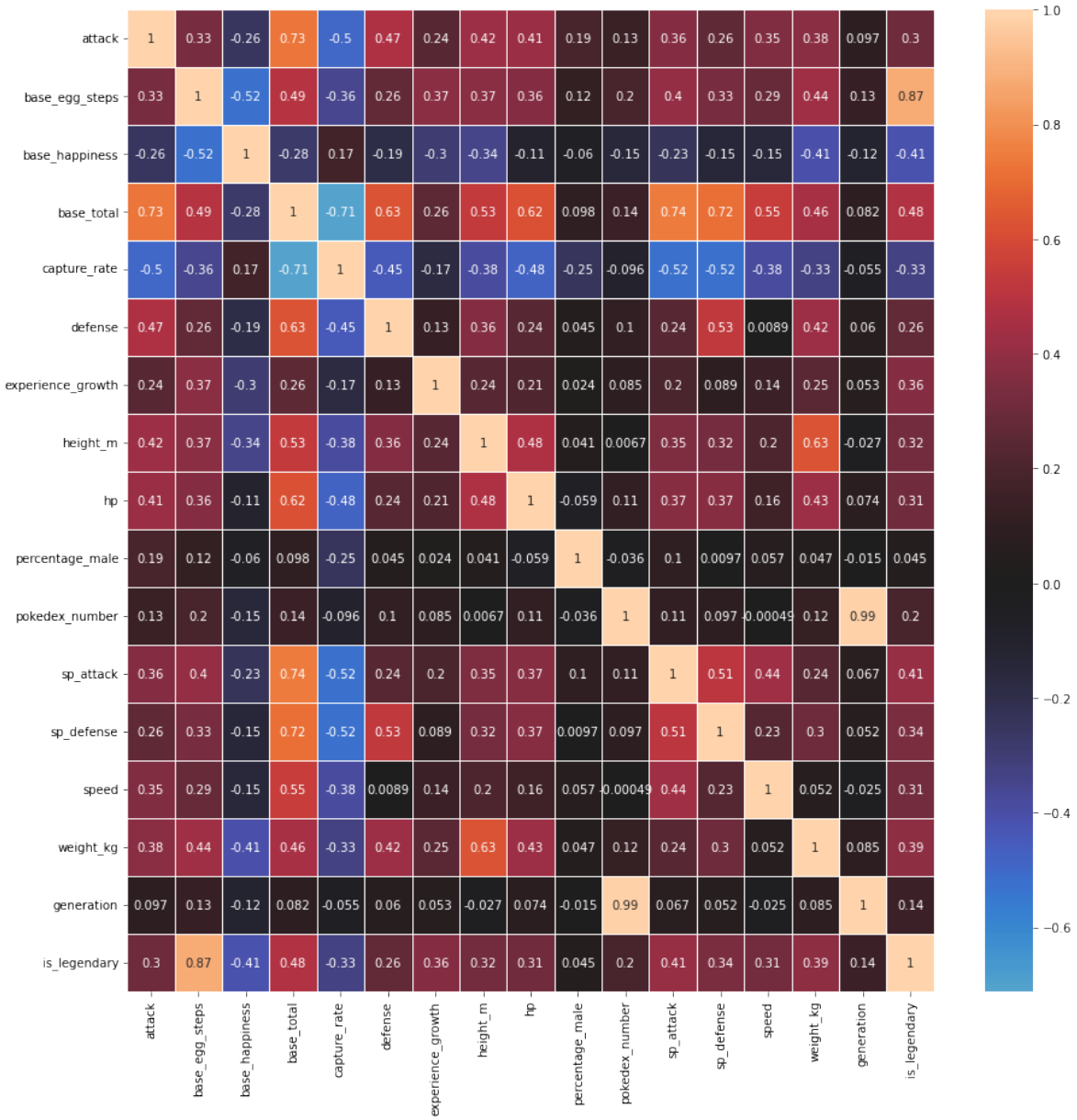


Figura 13: Correlació de les dades

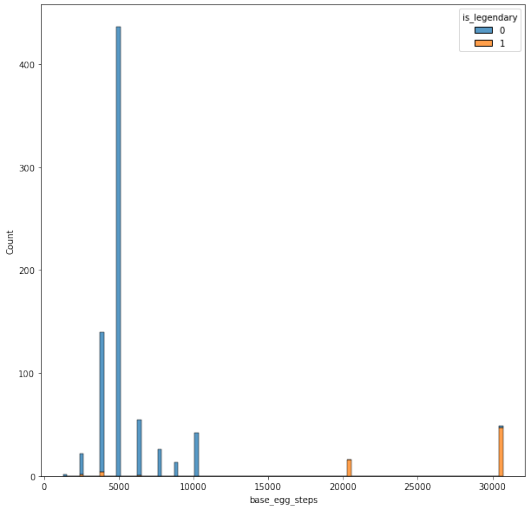


Figura 14: Passos per fer eclosió

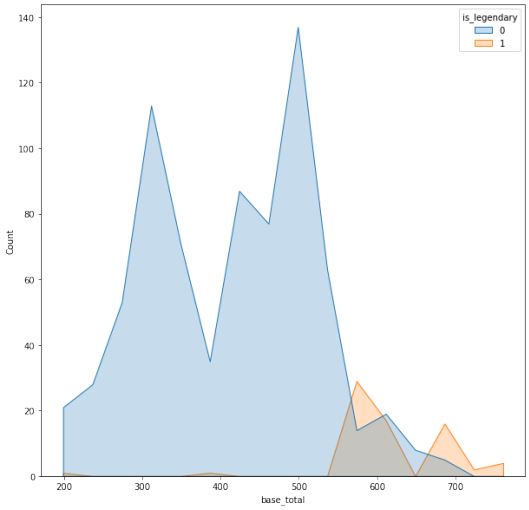


Figura 15: Base de poder

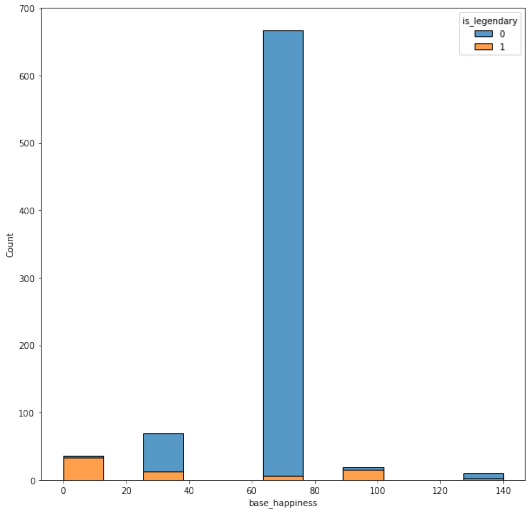


Figura 16: Felicitat base

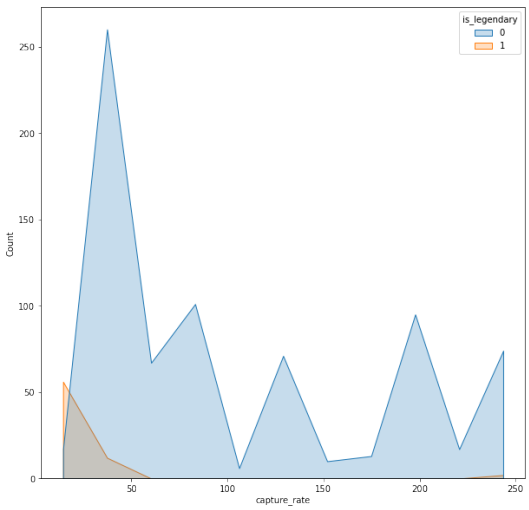


Figura 17: Rati de captura

2.2 Preprocessing (normalitzation, outlier removal, feature selection..)

- Estan les dades normalitzades? Caldria fer-ho?
  No, les dades no estan normalitzades: l'escala amb que es mesuren els atributs no és la mateixa per a tots. Pensem que serà convenient normalitzar les dades.

- En cas que les normalitzeu, quin tipus de normalització serà més adient per les vostres dades?
  Per assegurar-nos de que totes les dades estan normalitzades i no complicar-nos massa aplicarem a tots la mateixa normalització, el *StandardScaler* que transforma les dades perquè la mitja sigui 0 i la variància 1.

- Teniu gaires dades sense informació? Els *NaN*s a pandas? Tingueu en compte que hi ha mètodes que no els toleren durant el aprenentatge. Com afecta a la classificació si les filtrem? I si les reompliu? Com ho faríeu?
  Sí, tenim un atribut on és molt freqüent trobar *NaN*s: *percentage_male*. Aquests *NaN* tenen una explicació lògica: hi ha tot un conjunt de Pokémon que no tenen gènere (anomenats *genderless*). Hem aprofitat la presència d'aquests *NaN* per a crear un atribut *genderless*, que val 1 per als casos on *percentage_male* és *NaN*, i zero altrament. Aquesta variable sembla tenir força correlació amb el comportament de la variable objectiu *is_legendary*. Un altre atribut que té *NaN* és *type_2*, ja que hi ha força pokémon que són d'un sol tipus. Hem provat a crear un atribut

per a els Pokémon d'un sol tipus, però no té gaire correlació amb la variable objectiu. Els últims atributs amb *NaN* són *height_m* i *weight_kg* on apareixen els pokémons que tenen una forma alola alternativa. Aquest atribut té el valor *NaN* en aquests casos inclús quan les dades de les formes coincideixen.

Figura 18: Correlació de noves dades

- Teniu dades categoriques? Quina seria la codificació amb més sentit?
  Sí, tenim dades categòriques als atributs *type*1, *type*2, *classfication* i *abilities*. Les variable *classfication* i *abilities* no les farem servir ja que hi ha una quantitat excessiva (per diverses mostres la seva categoria és única, per tant, el regressor ignoraria aquestes variables al training igualment) i no sembla aportar-nos gaire informació, i com que *type*2 té força *NaN* (que indican que no té un segon tipus) i representa el mateix que *type*1 l'ajuntarem amb aquesta per fer la codificació. Per tant, per a aquesta combinació, provarem a fer ús d'una codificació 'OneHotEncoder', ja que 'OrdinalEncoder' pot donar problemes (trobar relacions espúries entre l'ordinal que representa cada categoria amb el comportament de la variable objectiu).

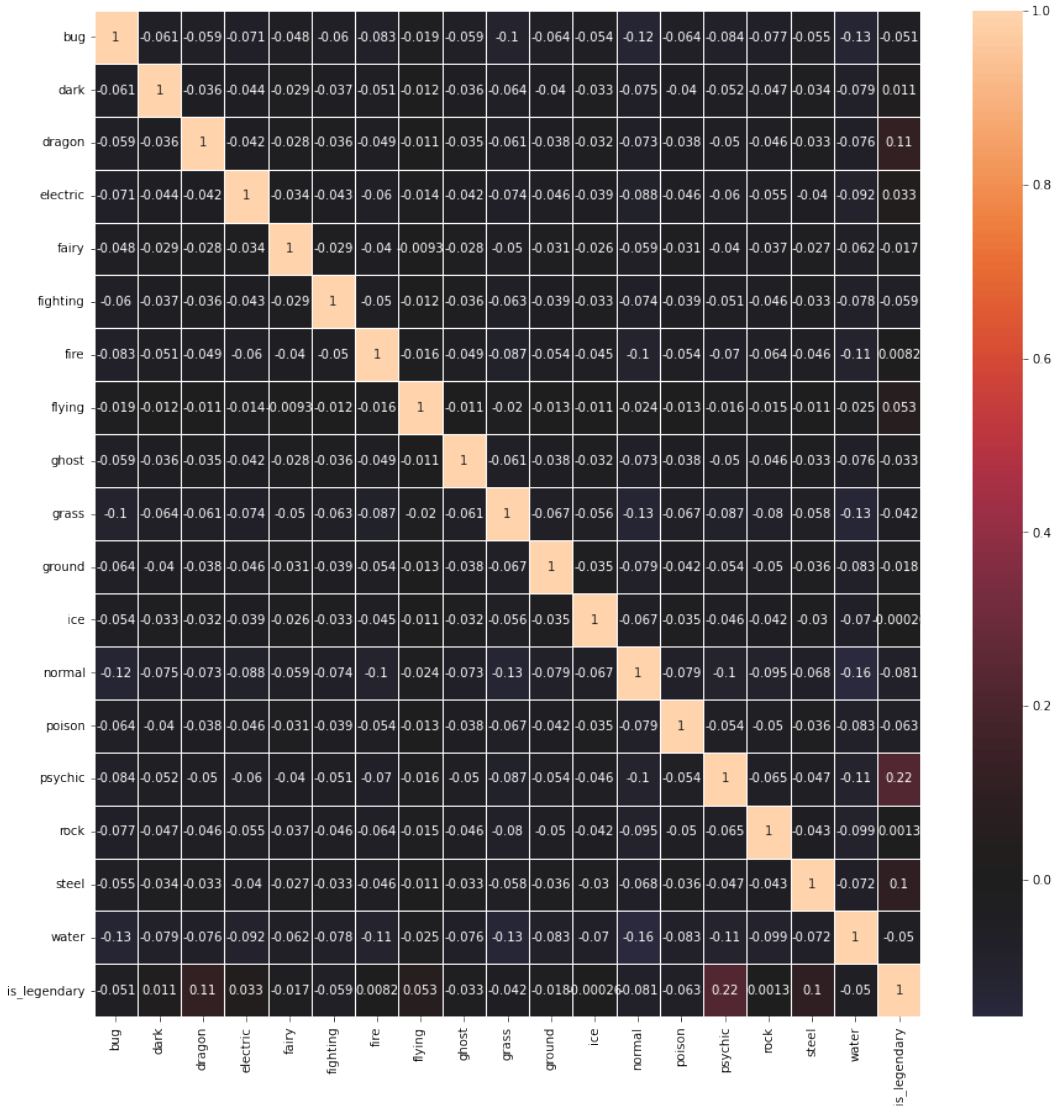


Figura 19: Tipus

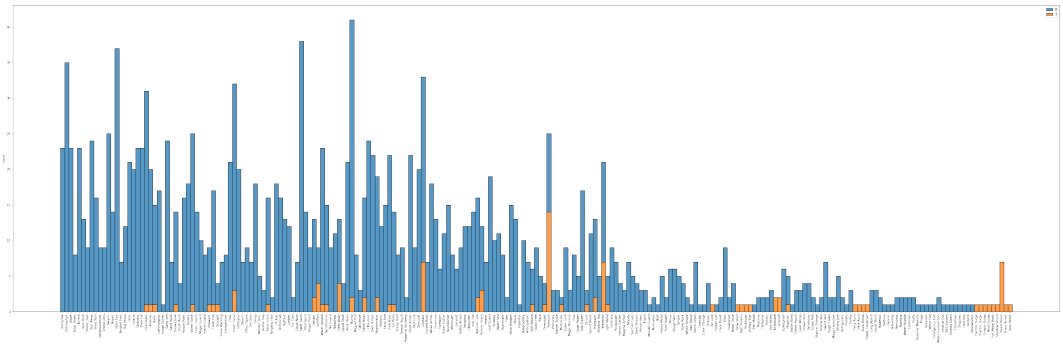


Figura 20: Habilitats

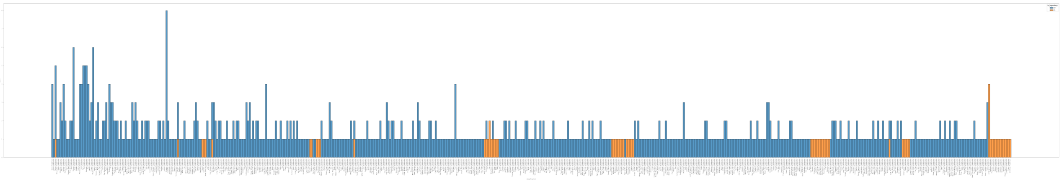


Figura 21: Classificació

- Caldria aplicar `sklearn.decomposition.PCA` ? Quins beneficis o inconvenients trobaríeu?
  No, només usant 3 bons atributs ja podem fer classificacions quasi perfectes (com es veu més endevant) aixi que no val la pena fer-ho.

2.3 Model Selection

- Quins models heu considerat?
  Logístic regression, SVM amb kernels linears i kernels RBF.
- Considereu les SVM amb els diferents kernels implementats.
  Si, el provem amb kernels linears i RBF.
- Quin creieu que serà el més precís?
  El més precís és el SVM amb kernel RBF
- Quin serà el més ràpid?
  El model més ràpid és el Logístic regression.
- Seria una bona idea fer un 'ensemble'? Quins inconvenients creieu que pot haver-hi? Els resultats ja son prou bons, per tant no sembla necessari fer un 'ensemble', ja que seria més lent i difícilment ho milloraria d'una manera significativa.

Model	Percentatge	Mitjana general		Mitjana ponderada	
		precisió	f1 score	precisió	f1 score
Logístic	50%	0.96	0.97	0.99	0.99
	70%	0.99	0.96	0.99	0.99
	80%	0.95	0.98	0.94	0.94
SVM Lineal	50%	0.96	0.96	0.99	0.99
	70%	1.00	0.99	1.00	1.00
	80%	0.94	0.96	0.99	0.99
SVM RBF	50%	0.97	0.97	0.99	0.99
	70%	0.97	0.96	0.99	0.99
	80%	1.00	1.00	1.00	1.00

Figura 22: Resultats dels models

2.4 Crossvalidation

- Per què és important cross-validar els resultats?
  Per comprobar que no s'està produint overfitting al fer la regressió del nostre model.
- Separa la base de dades en el conjunt de train-test. Com de fiables serán els resultats obtinguts? En quins casos serà més fiable, si tenim moltes dades d'entrenament o poques?
  Si tenim moltes dades d'entrenament i poques de cross-validation, el regressor tindrà major coneixement per generalitzar. En canvi si en té menys pot ser que no hi hagi casos que hagi pogut aprendre pero el test ens donarà més seguretat.

- Quin tipus de K-fold heu escollit? Quants conjunts heu seleccionat (quina k)? Com afecta els diferents valors de k?
  Hem provat per K d'entre 3 i 5:

Model	K	cross validation					Mitja
Logístic	3	1.00	0.98	0.96			0.98
	4	0.99	1.00	0.96	0.94		0.97
	5	0.99	0.99	0.99	0.98	0.95	0.98
SVM Lineal	3	0.99	0.96	0.96			0.97
	4	0.99	0.98	0.98	0.96		0.98
	5	0.98	0.99	0.99	0.96	0.93	0.97
SVM RBF	3	0.99	0.96	0.96			0.97
	4	0.99	0.98	0.98	0.96		0.98
	5	0.97	0.98	0.98	0.96	0.93	0.96

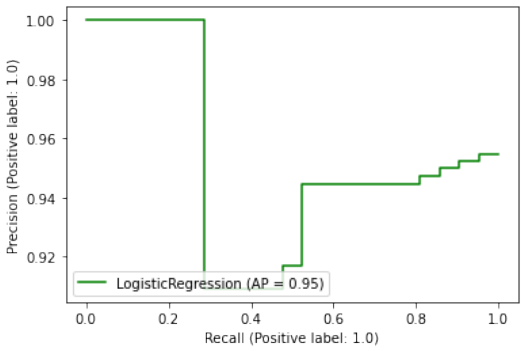
Figura 23: Resultats per diferents valors de K

- Es viable o convenient aplicar 'LeaveOneOut'?
  No es molt convenient ja que es tindria que recalcular el model 801 vegades i per tant trigaria molt de temps.

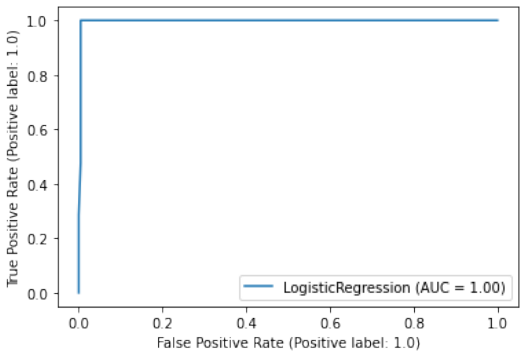
2.5 Metric Analysis

- A teoria, hem vist el resultat d'aplicar el 'accuracy_score' sobre dades no balancejades. Podrieu explicar i justificar quina de les següents mètriques serà la més adient pel vostre problema? 'accuracy_score', 'f1_score' o 'average_precision_score'
  Qualsevol d'aquestes ens pot indicar que tal ho està fent si la calculem per cada label i ajuntem aquests valors com una mitjana. (tal i com surt a la columna 'Mitjana general' de la taula dels models 22)

- Mostreu la Precisió-Recall Curve i la ROC Curve. Quina és més rellevant pel vostre dataset? Expliqueu amb les vostres paraules, la diferencia entre una i altre.

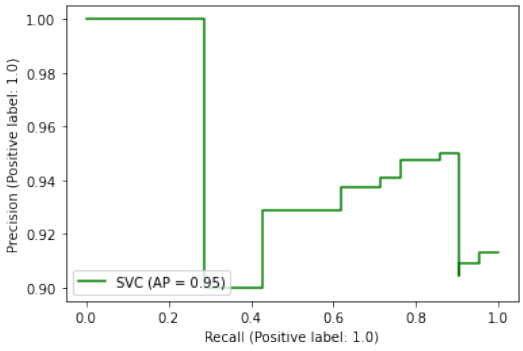


Corba Recall

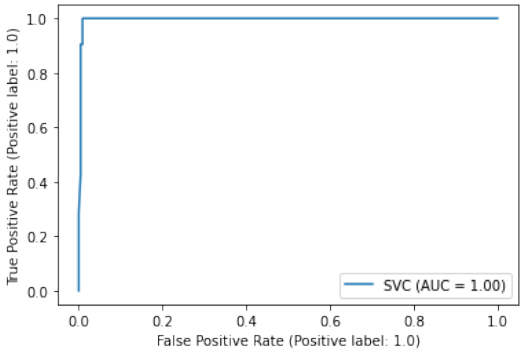


Corba ROC

Figura 24: Logístic

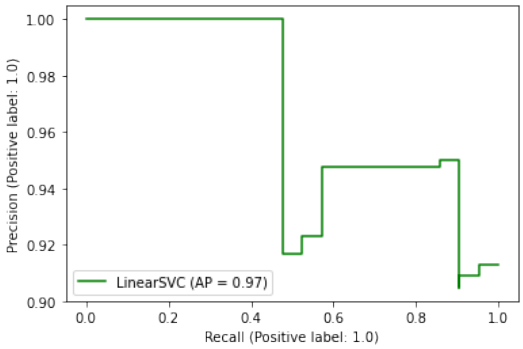


Corba Recall

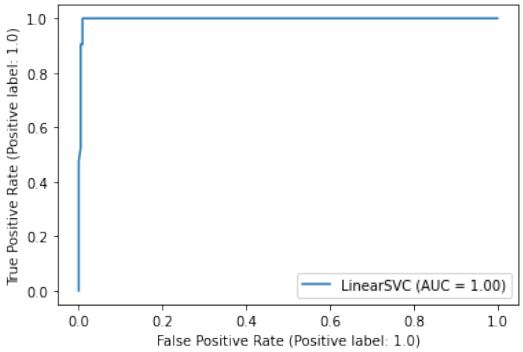


Corba ROC

Figura 25: SVM RBF



Corba Recall



Corba ROC

Figura 26: SVM Linear

La ROC Curve mostra el true positive rate contrastats amb el false positive rate. La precisió-Recall Curve mostra la precisió contrastada amb el recall. Mentres que el recall i el true positive rate son iguals, les corbes son diferents pq sifereixen el l'eix on estan situats aquests i el false positive rate no es igual a la precisió.

- Què mostra 'classification_report'? Quina métrica us fixareu per tal de optimitzar-ne la classificació pel vostre cas?

  Els resultats del 'classification_report' es troben a la taula dels models 22. D'aquesta taula el que més ens impora és la columna de 'Mitjana general', d'aquest columna utilitzarem el valor de 'f1 score'.

2.6 Hyperparameter Search

- Quines formes de buscar el millor paràmetre heu trobat? Són costoses computacionalment parlant?
  La primera és agafar diversos punts a l'atzar i anar probant, l'altra és fent una malla i examinant tots els punts d'aquesta.

- Si disposem de recursos limitats (per exemple, un PC durant 1 hora) quin dels dos mètodes creieu que obtindrà millor resultat final? Si tenim molts hiperparametres o l'execució per comprovar l'eficàcia és molt costosa, llavors serà millor agafar punts a l'atzar ja que agafes una mica de la idea general. Si no son molts o és calcula molt ràpid, serà millor fer una malla perquè el resultat es més fàcil de visualitzar.

- Existeixen altres mètodes de búsqueda més eficients?

- Feu la prova, i amb el model i el mètode de cross-validació escollit, configureu els diferents mètodes de búsqueda per a que s'executin durant el mateix temps (i.e. depenent del problema, 0,5∼1 hora). Analitzeu quin ha arribat a una millor solució. (estimeu el temps que trigarà a fer 1 training, i aixi trobeu el número de intents que podeu fer en cada cas)

Hem fet una proba pel model del regressor logístic per buscar quina era la C i la tolerància ideal pel mètode:

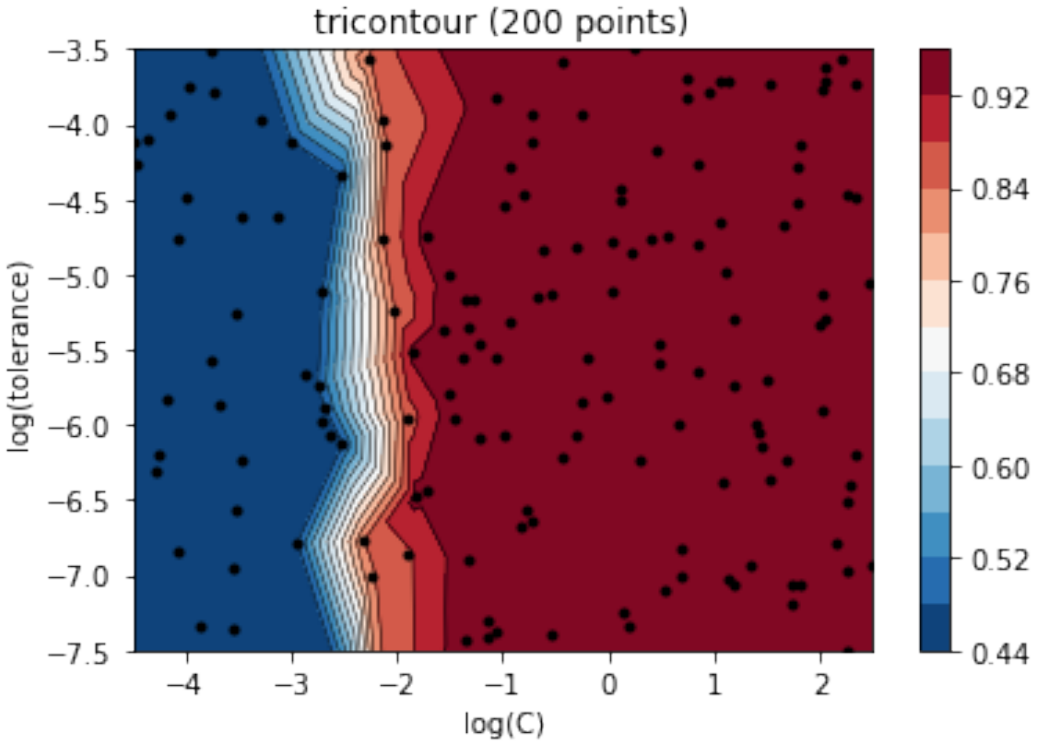


Figura 27: Busca d'hiperparametre amb 200 punts aleatoris

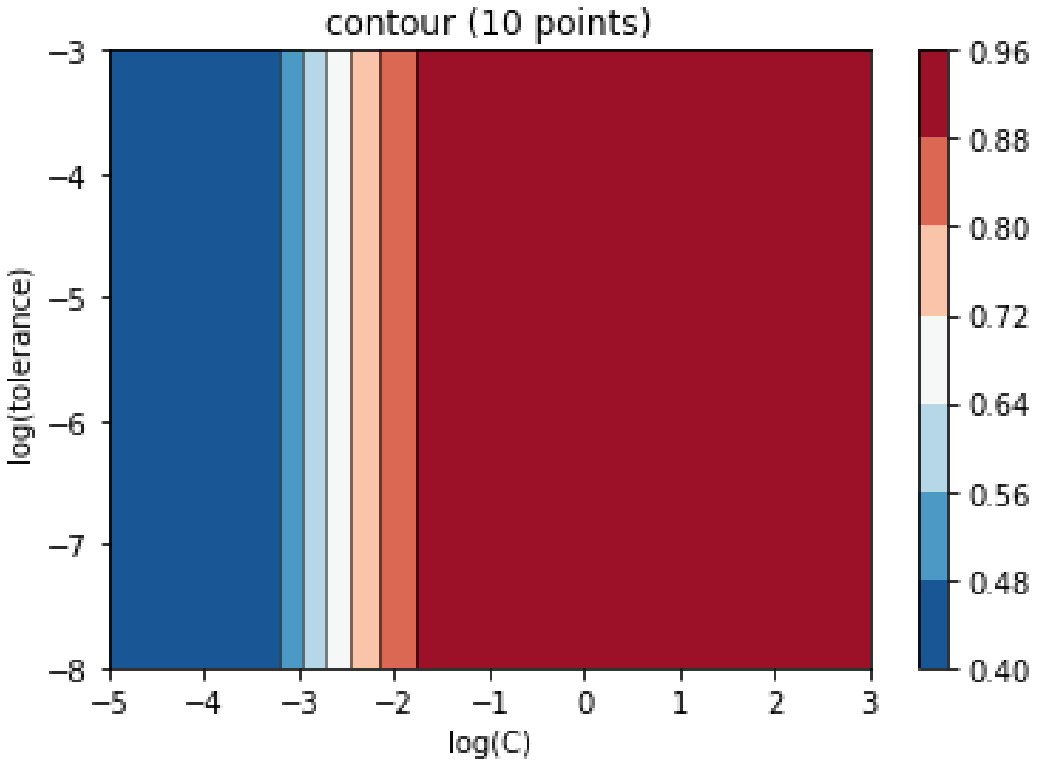


Figura 28: Busca d'hiperparametre amb una malla de 10×10 punts

Els resultats mostren que només sembla ser important tenir una C una mica elevada. ($\log_{10}(\mathrm{C}) > -1 \Rightarrow \mathrm{C} > 0.1$) Els resultats dels hiperparametres es poden veure més clarament al gràfic fet amb la malla.

Referències

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.