MLTOX Simone

Simone D'Ambrosi

April 2020

1 Intro

La tesi si basa sul lavoro precedentemente svolto dagli studenti dell'EPFL. Gli studi di dati tossicologoci rilevati sull'ambiente vengono usati per trarre informazione sugli agenti chimici e il loro effetto tossico sull'ambiente, in particolare sugli animali.

Lo scopo di questo lavoro consiste nel concentrarsi sull'ambiente acquatico e predirre le potenzialità tossiche di composti non testati su specie di animali già testate e viceversa. Questo è reso possibile attraverso tecniche di Machine Learning quali K-NN e Random Forest. La metrica per la valutazione dei modelli è l'accuratezza.

I dati raccolti sono stati elaborati, riordinati e processati dagli studenti edll'EPFL e, quindi pronti per utilizzarli nella fase di creazione di modelli.

Inizialmente, i dati raccolti e forniti sono stati filtrati per gli endopoint presi in considerazione LC50 e EC50, ossia si è fissata la soglia di concentrazione dei composti a 1mg/L (poi divisa in più classi) ritenuta sufficiente per uccidere metà della popolazione. Altri filtri riguardano la natura degli animali testati: si considerano solo i Pesci.

Dal CAS number dei composti si sono estratti gli SMILES ossia una rappresentazione ASCII dei composti da cui si possono estrarre molte informazioni riguardo le caratteristiche degli agenti chimici. Dopo aver estratto queste informazioni si è passati al preprocessing delle variabili ritenute valide per l'analisi. Sono state applicate trasformazioni per rendere le variabili più regolari.

Un grande ed importante punto sono gli esperimenti ripetuti. Nei dati forniti sono presenti esperimenti in cui una stessa specie di pesce è stata sottoposta allo stesso agente chimico sotto le stesse condizioni, ma avendo risultati diversi. Affinchè si possa eliminare la soggettività degli esperimenti si è deciso di risolvere aggregando i risultati dei diversi esperimenti tramite la mediana della concentrazione. Un possibile sviluppo è quello di considerare una misura di *fiducia* che abbiamo negli esperimenti in quanto, se si hanno 5 esperimenti di cui 3 hanno l'effetto A e 2 l'effetto B, l'esperimento *aggregato* sarà meno affidabile rispetto all'aggregazione di 5 esperimenti di cui tutti e 5 hanno l'effetto A.

Da qui comincia la mia analisi.

In via preliminare, si è considerata la possibilità di aggiungere nuove caratteristiche indirettamente estraibili dal CAS number. Dopo aver verificato la prove-

nienza delle variabili preesistenti, l'unica caratteristica estratta dallo SMILES, aggiunta da me, è stata il conteggio dei gruppi OH presenti nei composti.

Un'altra fondamentale feature è il *Pubchem2d* ossia una serie di 881 caratteristiche ciascuna rappresentata da uno 0, che ne indica l'assenza, o 1, che ne indica la presenza.

Il Pubchem2d è un contenitore di molta informazione fondamentale per capire se l'utilizzo delle caratteristiche estratte dallo SMILES risulta essere una buona riduzione dimensionale per il nostro problema.

Utilizzando questa informazione (Pubchem2d), ho potuto sviluppare due nuove *distanze tra composti*:

- La prima utilizzando la distanza di Hamming che conta il numero di posizione in cui due stringhe differiscono.
- La seconda utilizzando l'indice di Tanimoto, che considera sia il numero di posizioni in cui differiscono sia come differiscono (se la stringa A ha 1 e B ha 0, allora si attribuisce "un punto" ad A).

I risultati delle tecniche di clustering consistono di suddividere in 3 macro-gruppi i composti per la distanza di Hamming, mentre per il Tanimoto Index è possibile considera anche 3 macro-gruppi ma sarebbe preferibile suddividerlo in 4, per un raggruppamento migliore.

La grande differenza tra le due distanza, al di là dei risultati, è ben visibile dagli istogrammi.

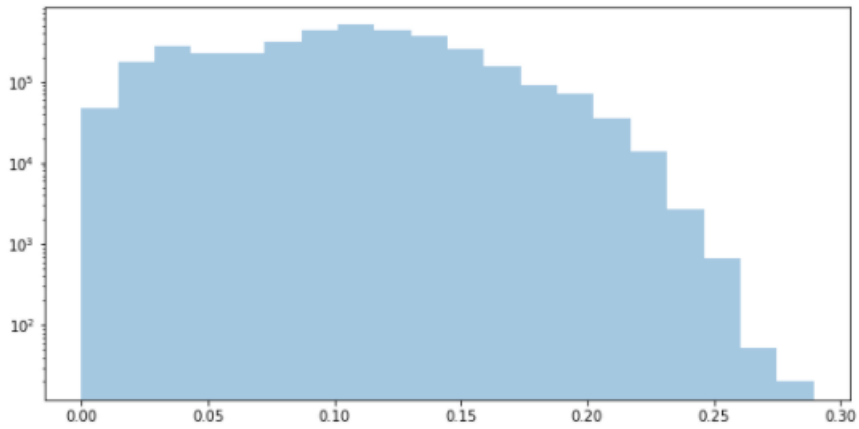

Figure 1: Hamming

da inserire grafici distanza con logbinning

Mentre Hamming varia tra 0 e 0.3 (anche se l'indice di natura varia tra 0 e 1), per Tanimoto varia tra 0 e 1. Di prassi, la distanza di Hamming andrebbe relativizzata tra 0 e 1, ma, per motivi di tempo (le operazioni con grandi matrici rubano molto tempo) non è stata effettuata, comunque i risultati futuri non vengono distorti eccessivamente.

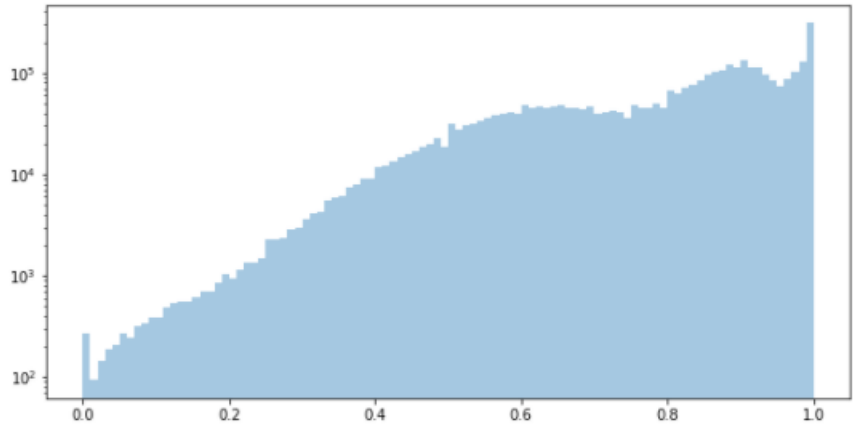

Figure 2: Tanimoto

Per avere un'idea dei cluster riporto i due dendrogrammi basati sulle due distanze:

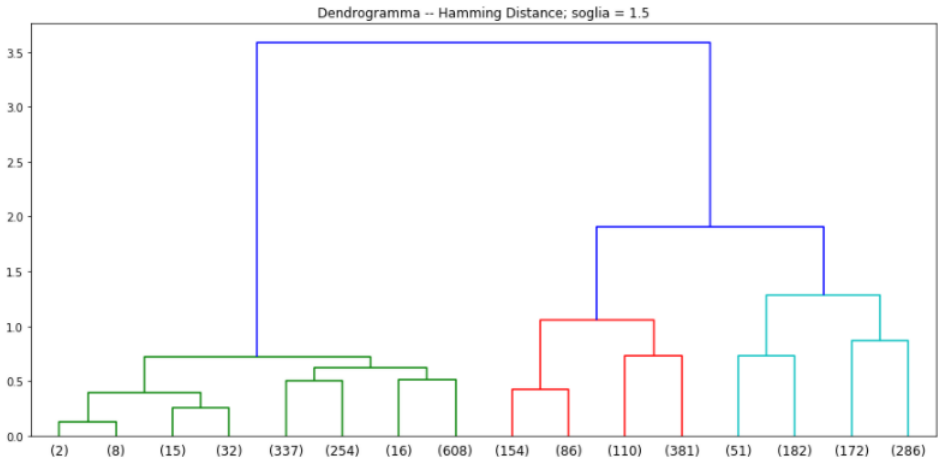

Figure 3: Hamming

Per confrontare i risultati delle due distanze ho portato il clustering basato su Tanimoto a produrre 3 cluster, quanti ne produceva Hamming basato su pubchem2d. L'adjusted Rand Index indica incorrelazione tra i due partizionamenti.

Un'analisi anch'essa preliminare è stata effettuata sugli animali testati, per capire se effettivamente le specie di pesci testate costituiscono un albero filogenetico oppure no. Nel file originario delle specie testate risulta esserci un problema di questo tipo, mentre, il dataset utilizzato per i modelli non presenta problemi (ovviamente :)). Questo imprevisto può essere risolto semplicemente

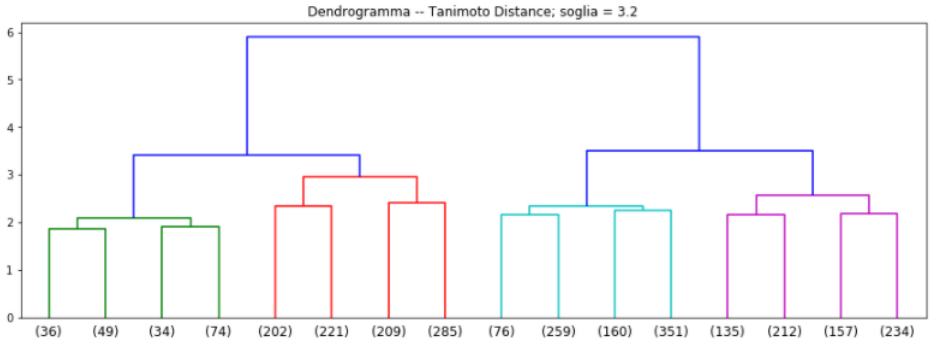

Figure 4: Tanimoto

considerando il *genere* e la *specie* unite, mantenendo comunque l'informazione derivante dal singolo *genere*.

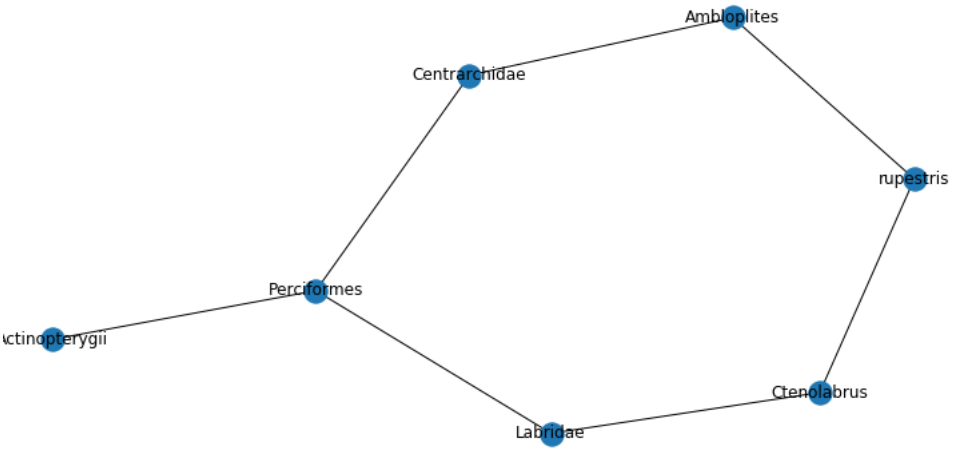

Figure 5: Problema albero filogenetico

Una rappresentazione dell'albero delle specie di pesci è fornito di seguito (ma può essere anche migliorato).

Figure 6: Albero filogenetico da migliorare

A questo punto, ci siamo spostati alla costruzione dei modelli di riferimento: KNN e Random Forest.

Una volta raccolti tutti i dati sviluppati, questi sono stati aggregati ai precedenti dati. In sostanza, ai dati precedentemente processati ho aggiunto l'informazione dei gruppi OH e quella derivante dai Pubchem2d. Il numero di gruppi OH è da considerare una variabile categorica ordinale.

1.1 Random Forest

É stato utilizzato il pacchetto **h2o**, che permette una gestione più oculata delle variabili categoriche. I risultati sono molto soddisfacenti. Sia per il caso binario che per il caso multiclasse l'encoding ottimale delle variabili categoriche è il cosidetto one-hot. La metrica di riferimento in questo caso è stata l'RMSE.

Per trovare la miglior combinazione dei parametri della random forest ho iniziato con la *sintonizzazione* della profondità massima degli alberi, che ha evidenziato il range di valori $[10, 20]$, sia per il caso binario che per il caso multiclasse.

A questo punto sono passato alla ricerca degli altri parametri, adottando una ricerca a griglia stocastica, ossia in un set di combinazioni dei parametri, la griglia ne sceglie un centinaio casualmente e li analizza. Se siamo fortunati si trova un modello buono.

I risultati del lavoro precedente sono:

		Accuracy	RMSE
k-NN	Binary	0.903	0.312
	Multiclass	0.740	0.687
MF	Binary	0.906	0.272
	Multiclass	0.662	0.637

TABLE VII
BEST RESULTS FOR OUR ML ALGORITHMS

Figure 7: Risultati lavoro precedente

I risultati correnti sono: per il caso binario, si raggiungono RMSE 0.2906 e accuratezza 0.9031, in linea con i risultati precedenti. Si è valutata anche le capacità di adattamento del modello deteriorandolo e cambiando split. I risultati sono comunque molto buoni; per il caso multiclasse, guardando la confusion matrix si può ricavare l'accuratezza che risulta pari a 0.7486 ed un RMSE pari a 0.5416, migliorando i risultati precedenti.

1.2 K-NN

Infine, ci siamo concentrati sul K-nearest neighbors, basato su matrici di distanza pre-costruite.

Il lavoro precedente ha usato la seguente formulazione: per le variabili categoriche si è optato per un encoding numerico in quanto il one-ho encoding portava l'aggiunta di 899 variabili e sapendo che per il KNN alte dimensionalità portano alla curse of dimensionality, si è cambiata strategia. Sulle variabili categoriche così processate si è calcolata una matrice di distanza con metrica Hamming ($d_1(i, j)$).

Sulle variabili continue semplicemente una matrice con metrica Euclidea ($d_2(i, j)$).

L'aggregazione delle due matrici è avvenuta tramite la seguente formulazione:

$$d(i,j) = \alpha d_1(i,j) + d_2(i,j)$$

Per raggiungere le migliori performance del modello, sono state fatte due selezioni: la prima sulle feature da usare che ha evidenziato l'utilizzo di tutte le caratteristiche a disposizione; la seconda riguarda gli iperparametri partendo dal numero di vicini k e poi α e un parametro denominato $leafsize$.

I risultati della Cross-Validation operata dai ragazzi del lavoro precedente ha evidenziato questi valori:

	α	K	Leaf size
Binary	$1.61 * 10^{-3}$	1	60
Multiclass	$4.28 * 10^{-3}$	1	10

TABLE IV
HYPER-PARAMETERS SELECTION FOR K-NN

Figure 8: risultati CV lavoro precedente

Alla luce della nuova caratteristica estratta, *Pubchem2d*, sono nate due nuove matrici di distanza, come detto in precedenza.

Sono state valutate prima singolarmente per capire le potenzialità dell'aggiunta alla matrice precedente.

I risultati sono i seguenti: per il knn del lavoro precedente si è fatto il tuning del parametro α per ogni k fissato:

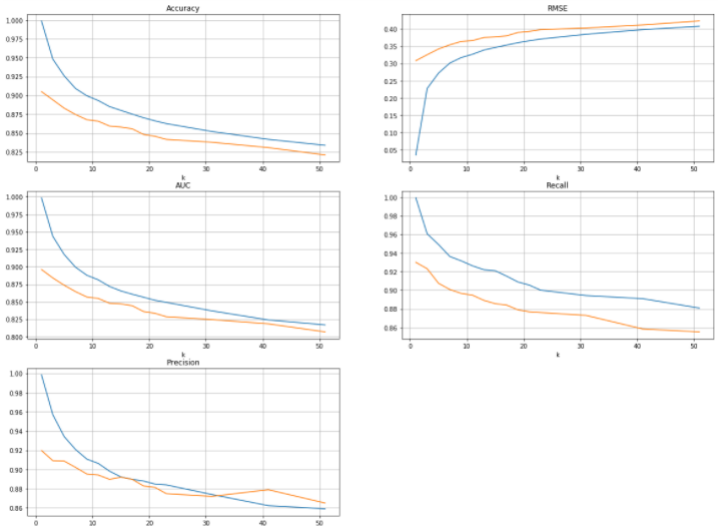

Figure 9: KNN lavoro precedente – Binary

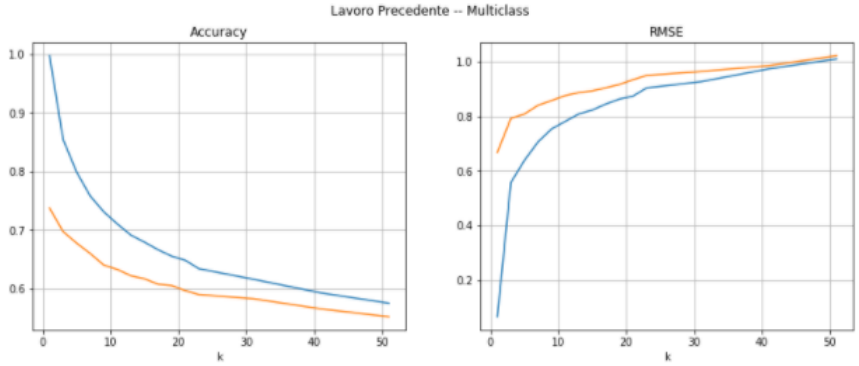

Figure 10: KNN lavoro precedente – Multiclass

che ha evidenziato, in accordo con quanto fatto precedentemente, $k = 1$ risulta essere ottimale e man mano che k cresce il parametro alpha varia di conseguenza, ma comunque il modello si degrada, sia nel caso binario che in quello multiclasse.

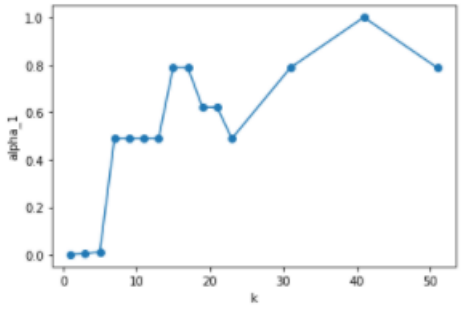

Figure 11: KNN Binary

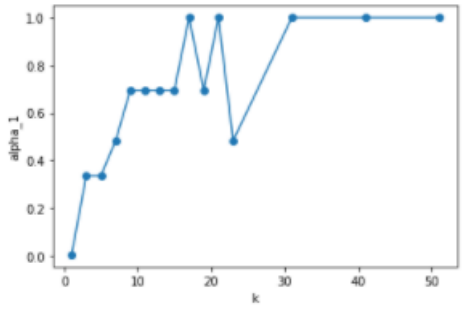

Figure 12: KNN multiclass

Dopo aver cross-validato il miglior modello ottenuto dal risultato precedente, siamo passati ad analizzare il KNN usando i pubchem2d con matrice di distanza con metrica Hamming:

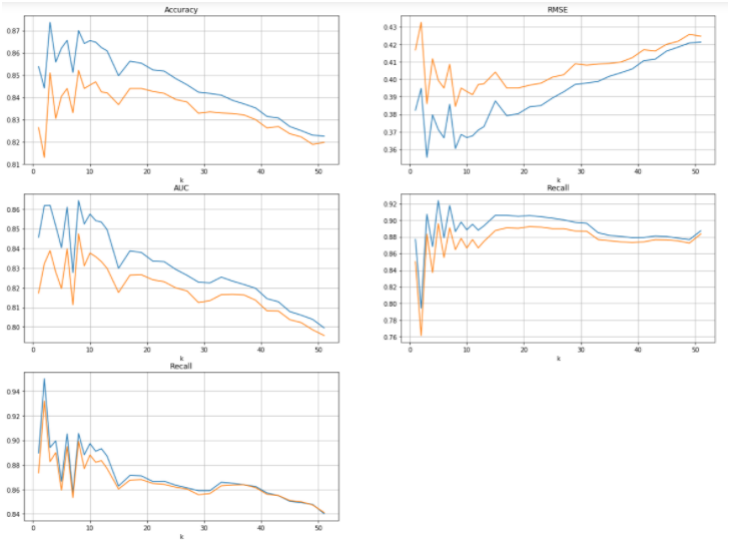

Figure 13: KNN Pubchem2d Hamming – Binary

Per il KNN Pubchem2d basato su Hamming si evidenziano più valori di k. Le oscillazioni sono dovute all'utilizzo di un valore per k pari, e con ciò la decisione del KNN in caso di pareggio consiste nell'utilizzare la modalità della variabile target più frequente.

Nel caso multiclasse si hanno i seguenti risultati:

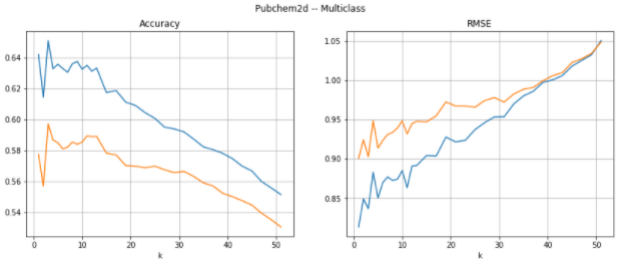

Figure 14: KNN Pubchem2d Hamming – Multiclasse

Infine sempre basandoci sui pubchem2d, utilizzando l'indice di tanimoto, abbiamo ottenuto questi risultati:

Successivamente, abbiamo deciso di non utilizzare la distanza ricavata dall'indice di Tanimoto in quanto ottiene un risultato non soddisfacente.

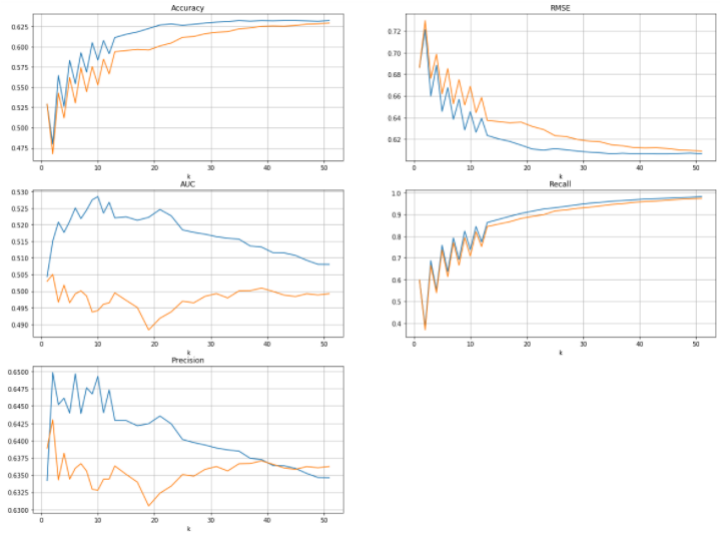

Figure 15: KNN Pubchem2d Tanimoto –Binary

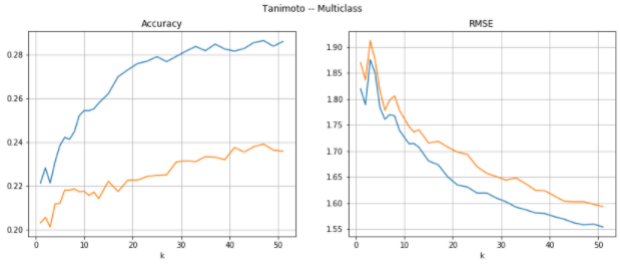

Figure 16: KNN Pubchem2d Tanimoto – Multiclasse

D'altra parte, invece, la distanza ricavata dai Pubchem2d tramite Hamming risulta essere degna di analisi più approfondite.

Questa distanza è stata aggregata alla precedente tramite la formula:

$$d(i,j) = \alpha_0 d_1(i,j) + d_2(i,j) + \alpha_1 d_3(i,j)$$

Dove $d_3(i,j)$ risulta essere la distanza ricavata dai Pubchem2d tramite Hamming. I due iperparametri α_0 e α_1 possono essere intesi rispettivamente come importanza delle variabili categoriche e dell'informazione dei pubchem2d rapportate alle variabili continue.

Il lavoro precedente può essere ottenuto ponendo $\alpha_1 = 0$. Partendo proprio da questo abbiamo cross-validato alternando l'ottimizzazione dei due parametri α_0 e α_1 per cercare di ottimizzare la metrica di riferimento, in questo caso l'accuratezza.

vari step dei parametri da inserire

I risultati di questa CV alternata sono compatibili a quelli precedenti, in media migliorandoli leggermente, non in maniera significativa. Possiamo dire questo perchè ad ogni step abbiamo cross-validato le metriche di riferimento.

Questa procedura è stata effettuata sia per il caso binario che il caso multi-classe.

Un'altra idea è stata quella di vedere se cambiando metrica di riferimento si raggiunge un risultato migliore in termini di accuratezza e di RMSE.

Impostando come metrica da ottimizzare la Log-Loss (o Cross-Entropy) definita come, nel caso binario:

$$-y \ln(p) + (1 - y) \ln(1 - p)$$

dove $y \in \{0, 1\}$ e $p = \mathbb{P}\{Y = 1\}$, si ottiene nel caso del lavoro precedente lo stesso risultato.

RASAR

Prossimo step è applicare il RASAR partendo dagli ultimi risultati sul KNN con Pubchem2d.