Elaborato

Corso: Analisi di dati biomedici

Riccardo La Grassa

Indice

Introduzione

Partendo dallo studio dell'articolo " Minimum spanning tree based one-class classifier " di Piotr Juszczak, David M.J. Tax, Elzbieta Pekalska, Robert P.W. Duin si è realizzato un classificatore one-class con una precisione soddisfacente su diversi tipi di dataset appartenenti a diverse famiglie (Nucleosome vs linker_elegans, Nucleosome vs linker_sapiens, Nucleosome vs linker_melanogaster, yeast and mouse). Per migliorare l'accuratezza dei risultati, si è optato per un metodo di validazione (cross validation) di tipo K-fold. In particolare, tra i risultati, vi sono diversi grafici derivati da diversi modelli addestrati su famiglie specifiche che mettono in evidenza, come all'aumentare della grandezza del training, si ha un aumento dell'errore dovuto ad uno stato della macchina in cui sta apprendendo troppo su dati specifici, questo fenomeno si chiama Overfitting. Sono stati scritti alcuni script in matlab che generano i risultati ottenuti, in particolare in fase di addestramento, si creerà un minimo albero di ricoprimento che muterà nel corso delle iterazioni inglobando le sequenze classificate correttamente rispettando diversi vincoli esposti successivamente in questo elaborato.

Estrazione caratteristiche e rappresentazione

Dato un alfabeto $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_r\}$ e $s = \sigma_{i1}, \sigma_{i2}, ..., \sigma_{iL} \in \Sigma^*$, si crea un vettore W che contiene tutte le disposizioni semplici con ripetizioni della cardinalità di $|\Sigma|^k$, dove k è scelto in arbitrio.
X_s è un vettore ottenuto calcolando la frequenza delle occorrenze di elementi di W in s.
Il vettore X_s verrà poi suddiviso elemento per elemento con il numero totale di coppie diverse prese a k a k nella stringa X_s. In tal modo, posso rappresentare una stringa in uno spazio numerico multidimensionale, ed applicare metriche per il calcolo della similarità tra stringhe.
Questa rappresentazione è chiamata k-meri.

Minimum Spanning tree

Sia G = (V, E) un grafo non orientato e connesso. Si definisce albero ricoprente di G un sottografo T ⊆ G tale che:

a) T è un albero;
b) T contiene tutti i vertici di G

Inoltre, si definisce costo dell'albero ricoprente di T ,
w(T) , la somma dei costi degli archi contenuti in T, ossia:

$$w(T) = \sum_{e \in t} w(e)$$

Date queste definizioni, l'obiettivo è quello di trovare il min(w(T)) fra tutti i possibili sottografi. Esistono diversi algoritmi per il calcolo, ai fini dell'elaborato è stato utilizzato l'algoritmo di Prim.
L'idea, pertanto, è quella di interconnettere tutte le sequenze genetiche, viste come nodi in una struttura multidimensionale, utilizzando come metrica la distanza euclidea per l'attribuzione dei pesi agli archi, e poter trovare successivamente il minimum spanning tree.
In generale, per due punti in uno spazio n-dimensionale, $P = (p_1, p_2, p_3, ..., p_n)$ e $Q = (q_1, q_2, q_3, ..., q_n)$ la distanza è calcolata come:

$$\sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

Matrice di adiacenza e metrica

Per ogni sequenza mappata su un piano multidimensionale, come già detto in precedenza, utilizzando la distanza euclidea, si costruirà la matrice di adiacenza contenente per ogni nodo i-esimo, la distanza con tutti i nodi j-esimo mappati sul piano.
Vale la seguente relazione:

$$M_{ij} = \begin{cases} 1 & \text{if } P_i \to P_j \\ 0 & \text{otherwise} \end{cases}$$

Nel nostro caso, inseriremo il peso dell'arco $w = P_i \Rightarrow P_j$
Si definisce matrice di adiacenza, una struttura del tipo:

$$A = \begin{pmatrix} 0 & w_{12} & \dots & w_{1n} \\ w_{21} & 0 & \dots & w_{2n} \\ \dots & \dots & 0 & \dots \\ w_{m1} & w_{m2} & \dots & 0 \end{pmatrix}$$

Dove, $w_{11} = w_{22} = \dots = w_{mn} = 0$, rappresentano i pesi lungo la diagonale. L'elaborato, tratta grafi non direzionati, quindi dalla precedente definizione si evince un'altra caratteristica:

$$[w_{ij}, \dots, w_{mn}]^T = [w_{ji}, \dots, w_{mn}]$$

pertanto, verrà a creare una triangolare superiore o inferiore in fase di elaborazione del tipo:

$$A = \begin{pmatrix} 0 & w_{12} & \dots & w_{1n} \\ 0 & 0 & \dots & w_{2n} \\ 0 & 0 & 0 & w_{m-1,n} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Richiamando l'istruzione sparse(A) di matlab, verranno eliminati tutti gli 0 per poter successivamente richiamare graphminspantree(A), che, applicando l'algoritmo di Prim (default, o Kruskal altrimenti) troverà il minimum spanning tree (mst).
In fase di addestramento del modello, si genera un vettore che contiene tutti i pesi degli archi dell' mst. In fase di apprendimento, al verificarsi di certe condizioni, il nodo classificato verrà inglobato nell'insieme del training, e la minima distanza tra nodo classificato e nodo dell' mst verrà aggiunta al vettore dei pesi e. Nello specifico, una volta addestrata la macchina sullo stesso set di dati (one class classifier), si manda in input il dataset della stessa famiglia contenente i linker e si ripete la procedura. In conclusione, si ricavano i risultati relativi alla sensività, specificità ed accuratezza della macchina. Partendo da sensività e 1-specificità ottenuti per k-meri = 2 con metrica euclidea si evince che i risultati migliori che utilizzando la metrica del coseno.

Threshold e mediana

In fase di apprendimento, rimane da calcolare la minima distanza fra la sequenza da classificare con ogni nodo dell'mst. Vale la seguente relazione:

$$\left(\min_{e_{ij} \in mst} d(x|e_{ij}) \right) \leq \theta$$

Nel nostro caso deriviamo il threshold theta, basandosi su una funzione di distribuzione degli archi pesati $w_{ij} = ||e_{ij}||$ in un dato mst;

Denotiamo $e = (||e_1||, ||e_2||, ..., ||e_n||)$ come una sequenza di scalari tali che:

$$||e_1|| \leq ||e_2|| \leq ... \leq ||e_n||$$

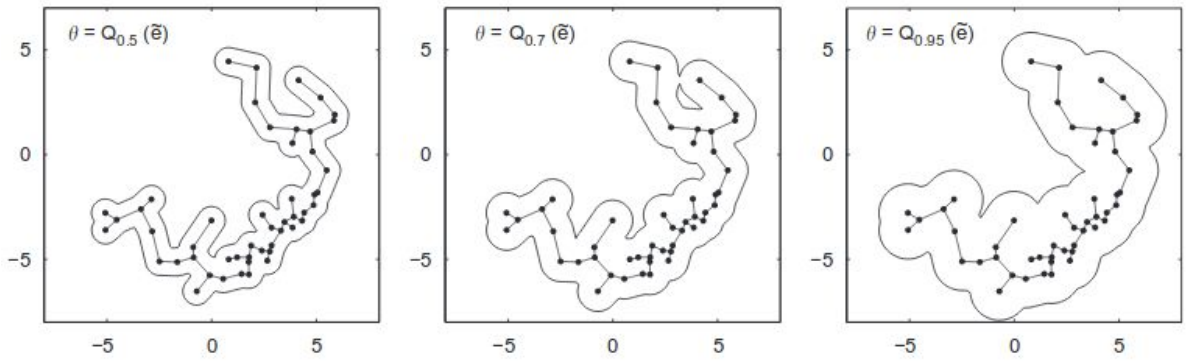
definiamo la funzione come:

$$Q_\alpha(e) = ||e_{\alpha n}||, \text{ con } \alpha \in [0, 1]$$

tale che:

$Q_0(e)$ rappresenta il minimo del vettore e
$Q_1(e)$ rappresenta il massimo del vettore e.

Il nostro valore del threshold scelto è $Q_{0.5}(e)$ che è uguale al calcolo della mediana del vettore e.
Le motivazioni per la quale non sia stata scelta la media sono per via dei possibili valori "anomali" che potrebbero essere presenti in e. Con il valore mediano, ricado esattamente al centro di questa distribuzione.



Le figure in alto, rappresentano la “frontiera” per la quale, sotto le condizioni elencate precedentemente, un nodo classificato debba appartenere o no all’mst. Si nota che, spostandosi verso la parte a valori più alti della distribuzione dei pesi di e la “frontiera” è più estesa, concedendo meno outlier in fase di classificazione rispetto al valore mediano, ma di conseguenza, un sequenza genetica con caratteristiche simili ma di un’altra famiglia ha maggiore probabilità di essere un falso positivo.

Coseno di similarità

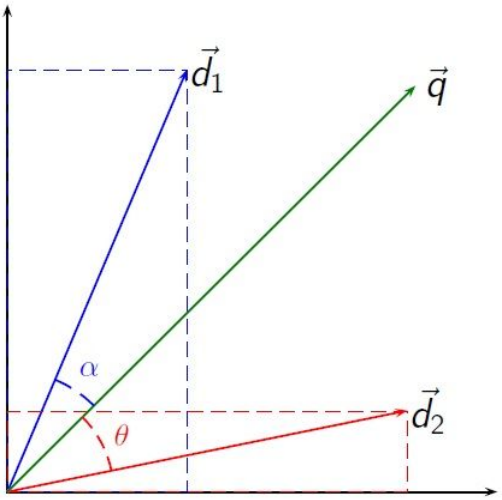
È stata utilizzata un’altra metrica per la comparazione dei risultati rispetto quelli con distanza euclidea, si tratta di una tecnica euristica per la misurazione della similitudine tra due vettori effettuata calcolando il coseno tra di loro.
Dati due vettori di attributi numerici, A e B, il livello di similarità tra di loro è espresso utilizzando la formula

$$\mathbf{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}.$$

o anche,

$$\frac{\sum_{k=1}^{n} A(k)B(k)}{\sqrt{\sum_{k=1}^{n} A(k)^2}\sqrt{\sum_{k=1}^{n} B(k)^2}}$$

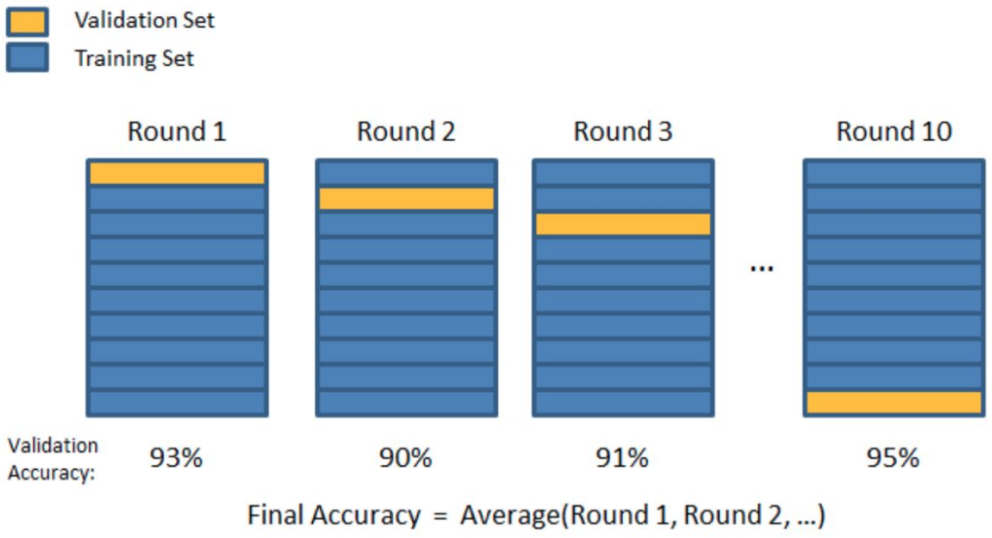
poiché le frequenze dei termini sono sempre valori positivi, si otterranno valori che vanno da 0 a +1, dove +1 indica la massima similarità (ma non necessariamente nello stesso ordine) e 0 il suo opposto.



L’immagine 3 vettori mappati su un piano bidimensionale e le loro reciproche distanze utilizzando la funzione coseno tra (d1,q) e (d2,q). In questo caso d1,q avrà una similarità più elevata rispetto l’altra coppia.

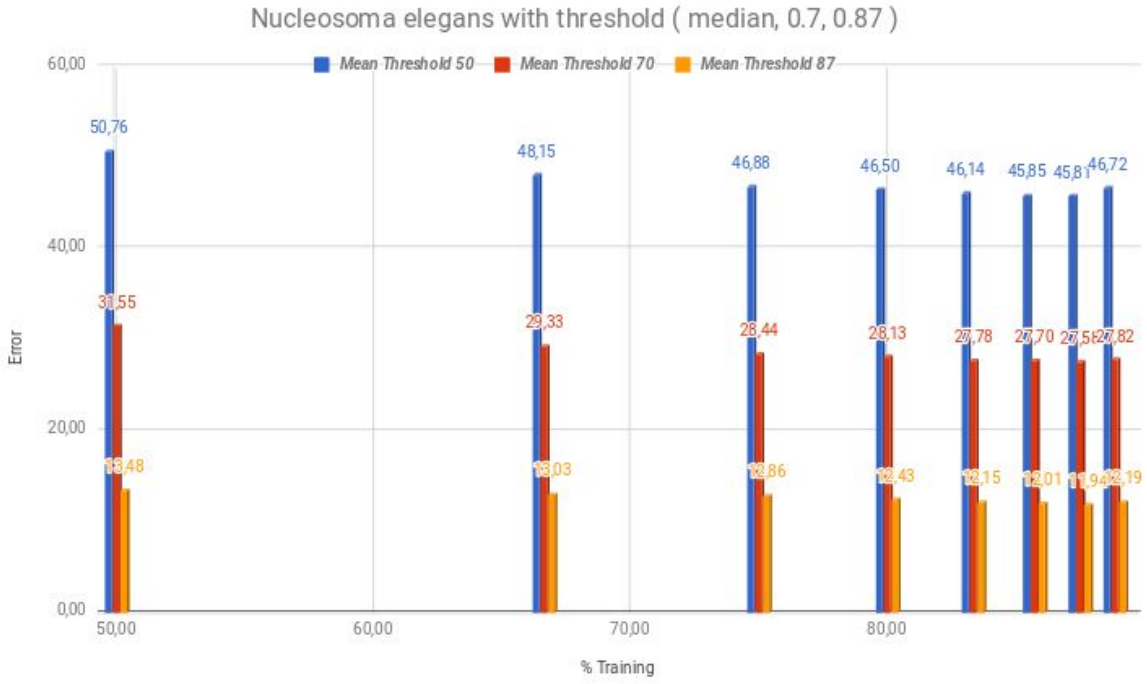
Commento dei risultati

È stato utilizzato un metodo di convalida incrociata chiamato k-fold, che consiste nella suddivisione del dataset totale in k parti di uguale numerosità e, ad ogni iterazione, la k-esima parte del dataset viene ad essere il *validation dataset*, mentre la restante parte costituisce il *training dataset*. Così, per ognuna delle k parti si allena il modello, evitando quindi problemi di overfitting, ma anche di campionamento asimmetrico del training dataset. In questo modo, si sceglie il modello addestrato che ha avuto un percentuale di accuratezza maggiore rispetto gli altri.

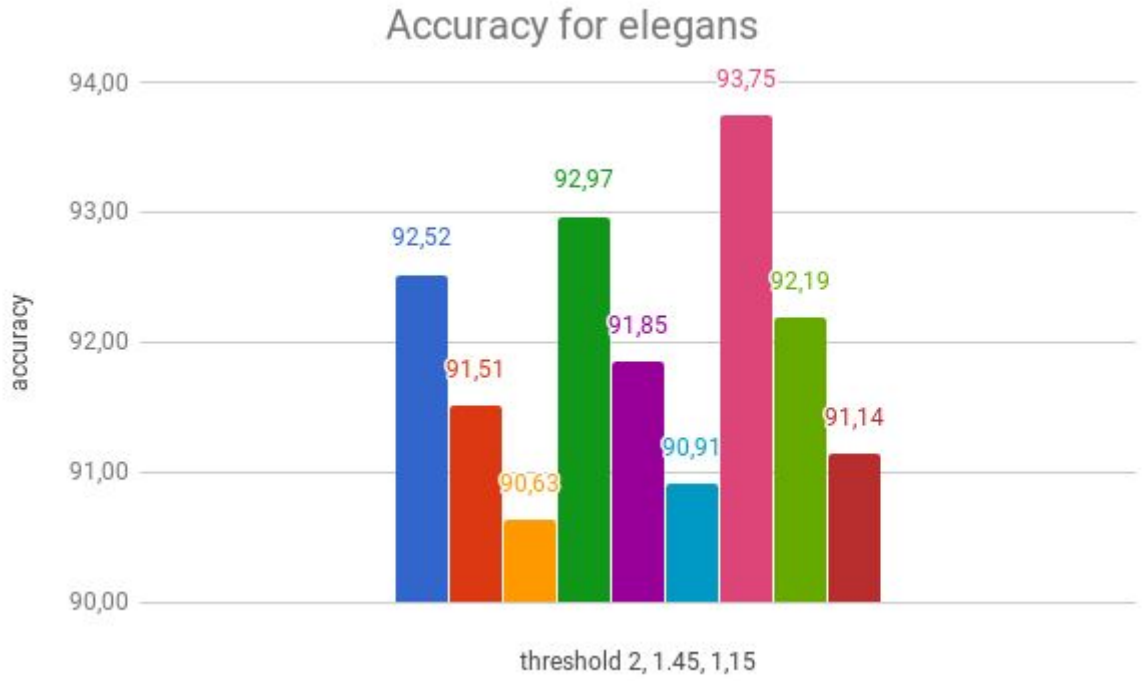


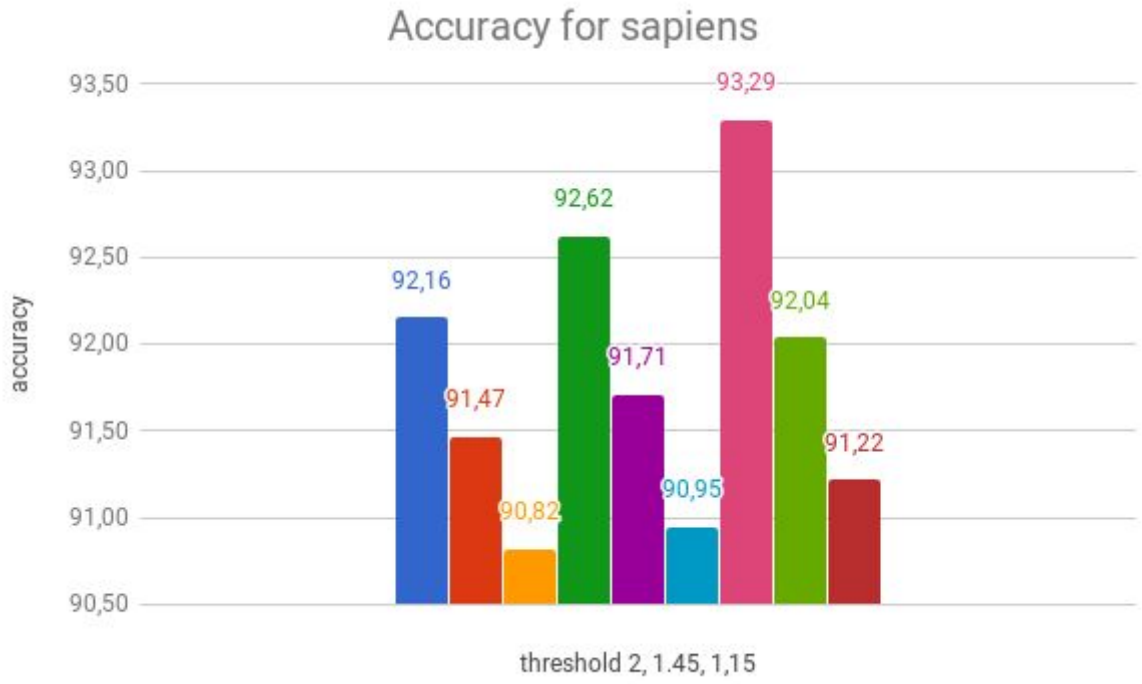
Risultati

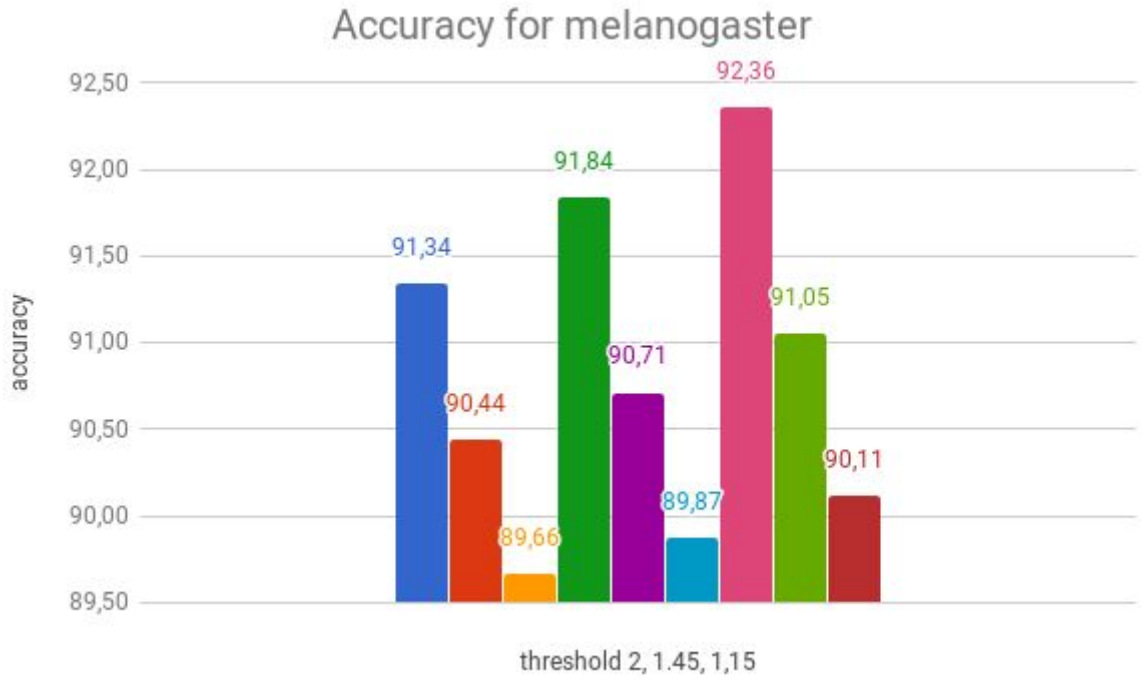
Come si evince dai risultati, comparandoli separatamente rispetto i loro threshold ed effettuando una media tra i kfold con lo stesso training size, il classificatore risulta avere un accuratezza maggiore del 92% con threshold 0.5 (mediana). Tuttavia, come dimostrano le immagini del paragrafo “Threshold e mediana” , l’aumento del threshold è proporzionale all'aumento di una falsa efficienza. Questo per via delle “frontiere” troppo espanse che causano una netta riduzione degli outlier, ottenendo come risultato dei falsi positivi. Di conseguenza, un threshold troppo basso diventerebbe restrittivo, con il risultato di escludere potenziali sequenze appartenenti alla classe.



In alto, un grafico a barre relativo al dataset dei nucleosomi elegans.
Si notano le differenti percentuali di errore (ossia nucleosomi correttamete etichettati diviso validation-set dei nucleosomi) in base al threshold scelto ed al variare della taglia del training di partenza.


Accuracy for elegans
94,00
93,00
92,00
91,00
90,00
accuracy
92,52
91,51
90,63
92,97
91,85
90,91
93,75
92,19
91,14
threshold 2, 1.45, 1,15

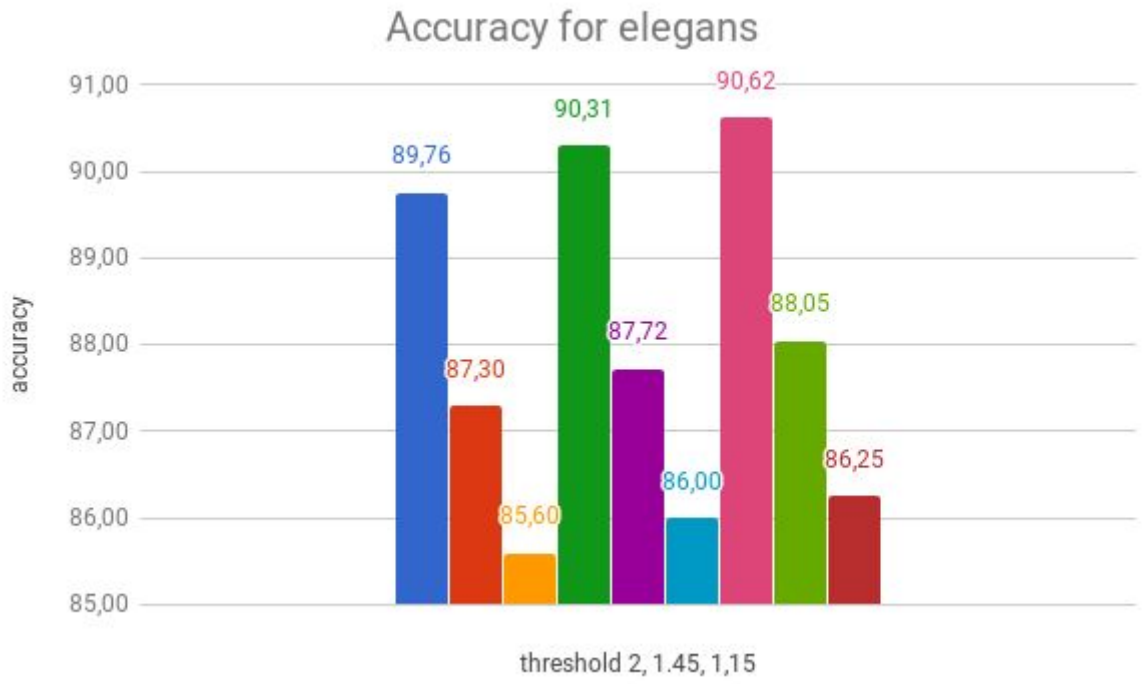

Accuracy for sapiens
93,50
93,00
92,50
92,00
91,50
91,00
90,50
accuracy
92,16
91,47
90,82
92,62
91,71
90,95
93,29
92,04
91,22
threshold 2, 1.45, 1,15

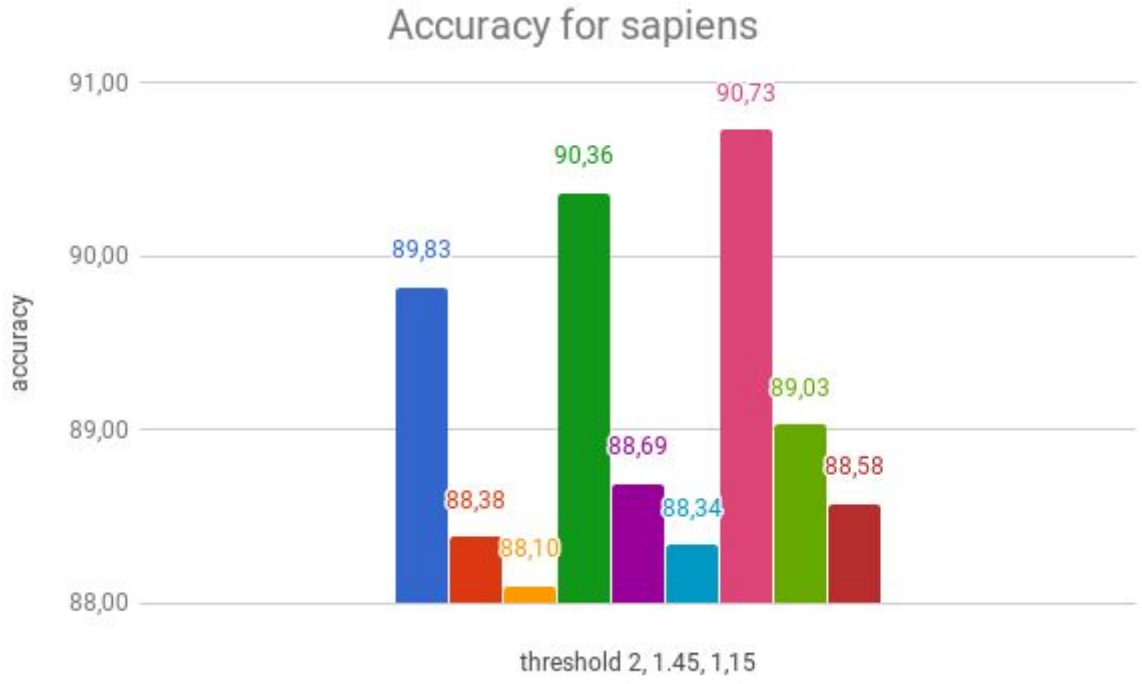


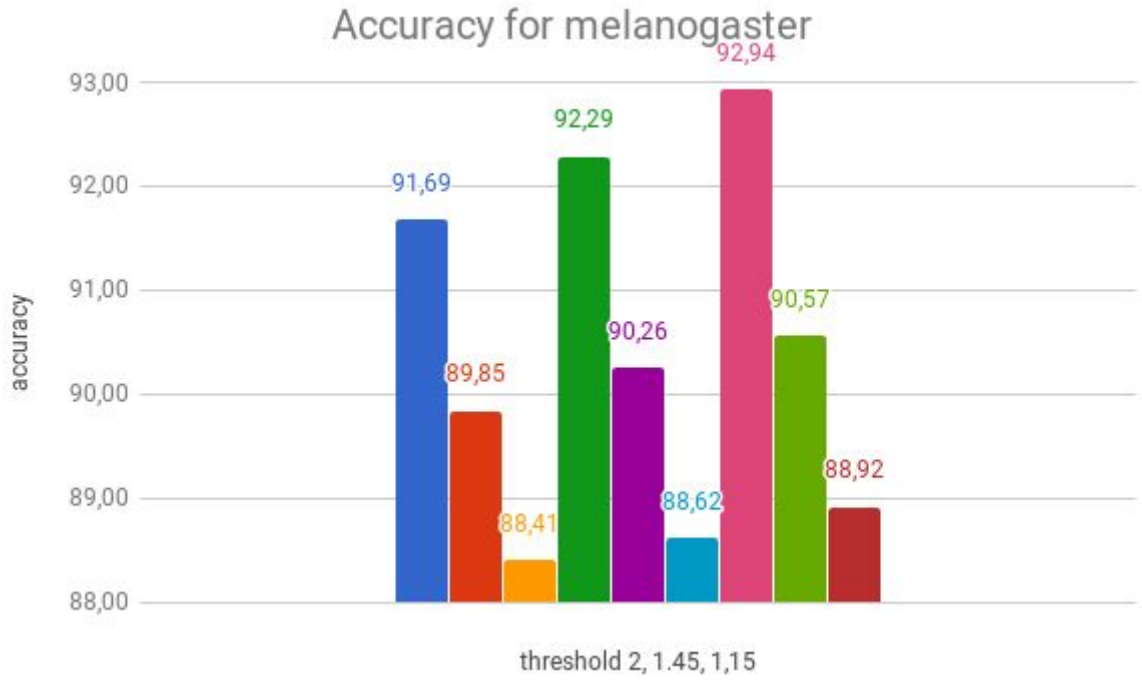
In alto sono rappresentate le relative accuratezze delle macchine addestrate al variare del training iniziale e del threshold (ogni gruppo di tre rappresenta l'accuratezza con il relativo threshold(2 1.45 1.15) e con training (85% 87% 88%))

In basso, troviamo i grafici considerando una rappresentazione k-meri >2.

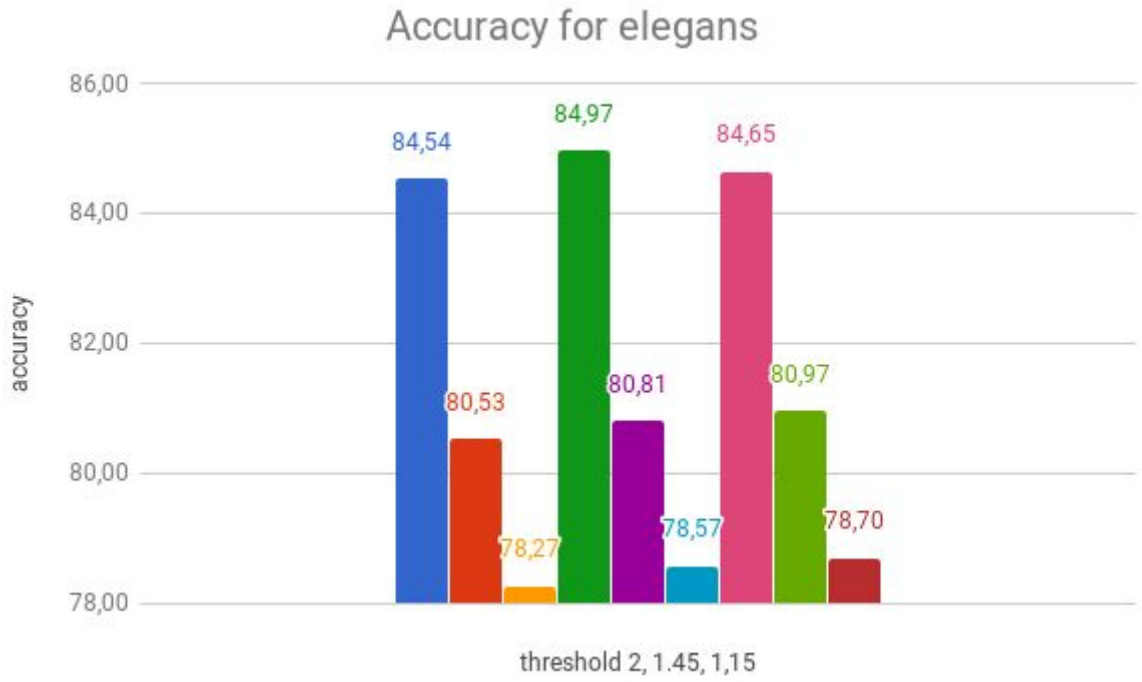
K-meri = 3

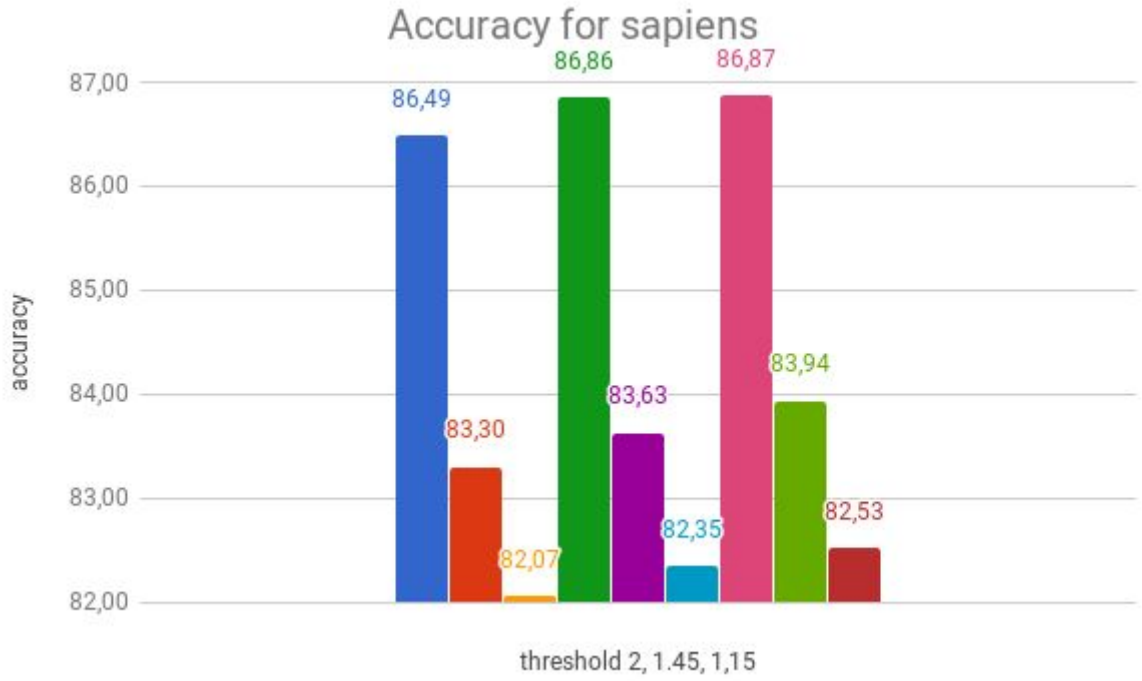

Accuracy for elegans
91,00
90,00
89,00
88,00
87,00
86,00
85,00
accuracy
89,76
87,30
85,60
90,31
87,72
86,00
90,62
88,05
86,25
threshold 2, 1.45, 1,15

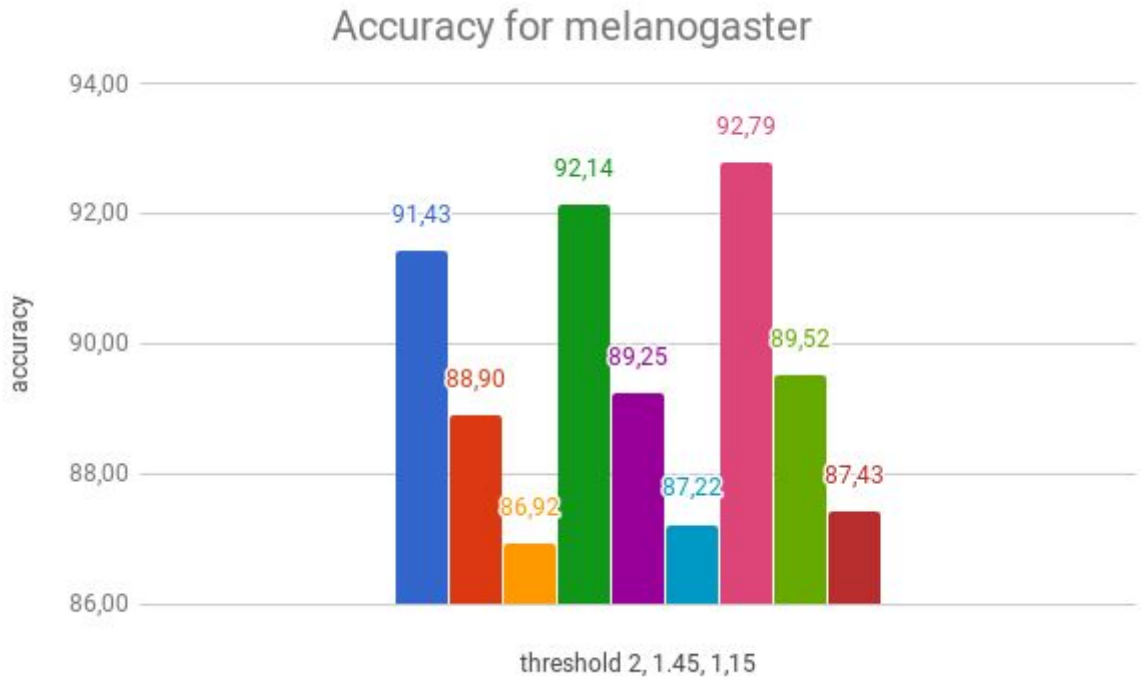

Accuracy for sapiens
91,00
90,00
89,00
88,00
accuracy
89,83
88,38
88,10
90,36
88,69
88,34
90,73
89,03
88,58
threshold 2, 1.45, 1,15


Accuracy for melanogaster
93,00
92,00
91,00
90,00
89,00
88,00
accuracy
91,69
89,85
88,41
92,29
90,26
88,62
92,94
90,57
88,92
threshold 2, 1.45, 1,15

K-meri=4

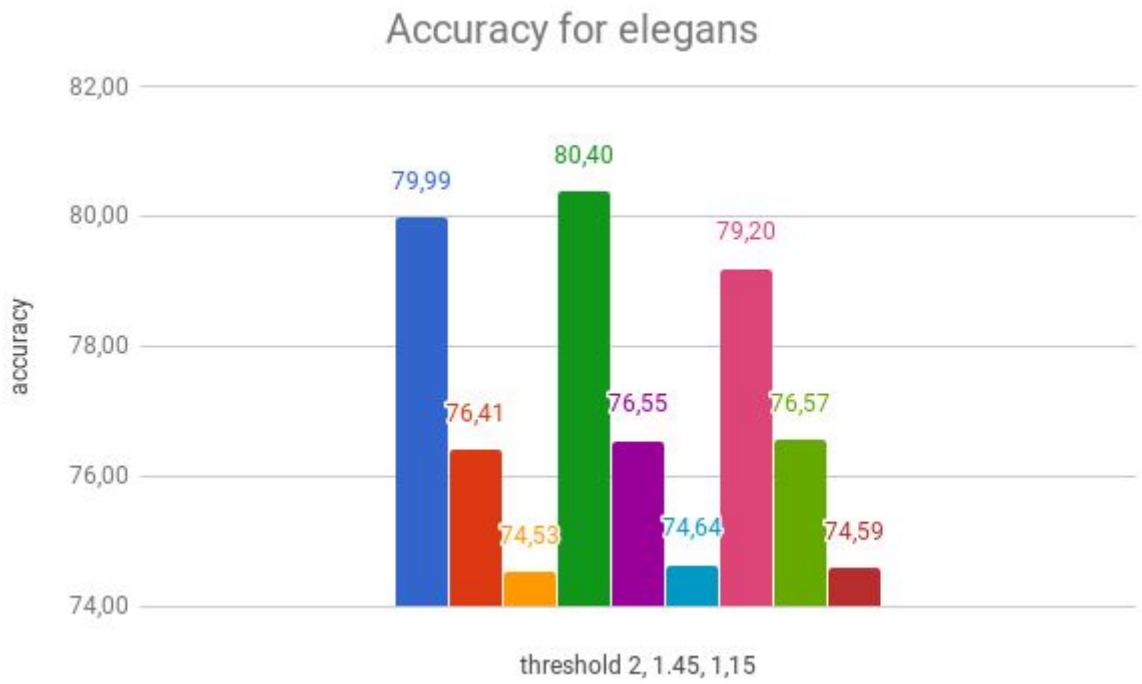

Accuracy for elegans
86,00
84,00
82,00
80,00
78,00
accuracy
84,54
80,53
78,27
84,97
80,81
78,57
84,65
80,97
78,70
threshold 2, 1.45, 1,15

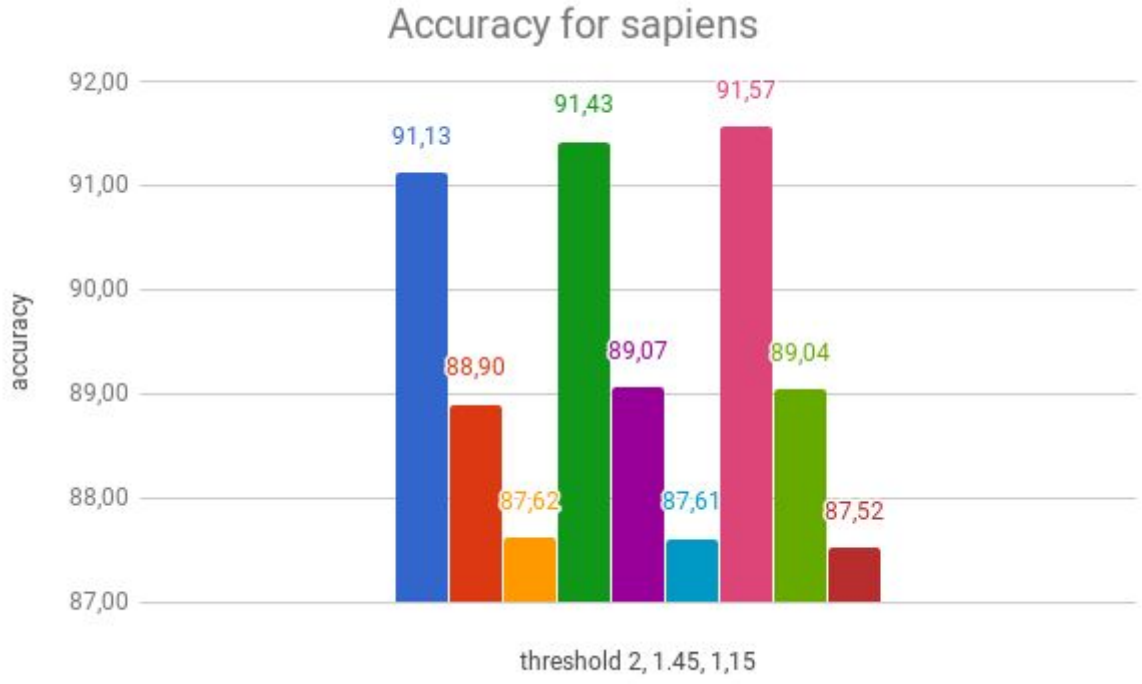

Accuracy for sapiens
87,00
86,00
85,00
84,00
83,00
82,00
accuracy
86,49
83,30
82,07
86,86
83,63
82,35
86,87
83,94
82,53
threshold 2, 1.45, 1,15

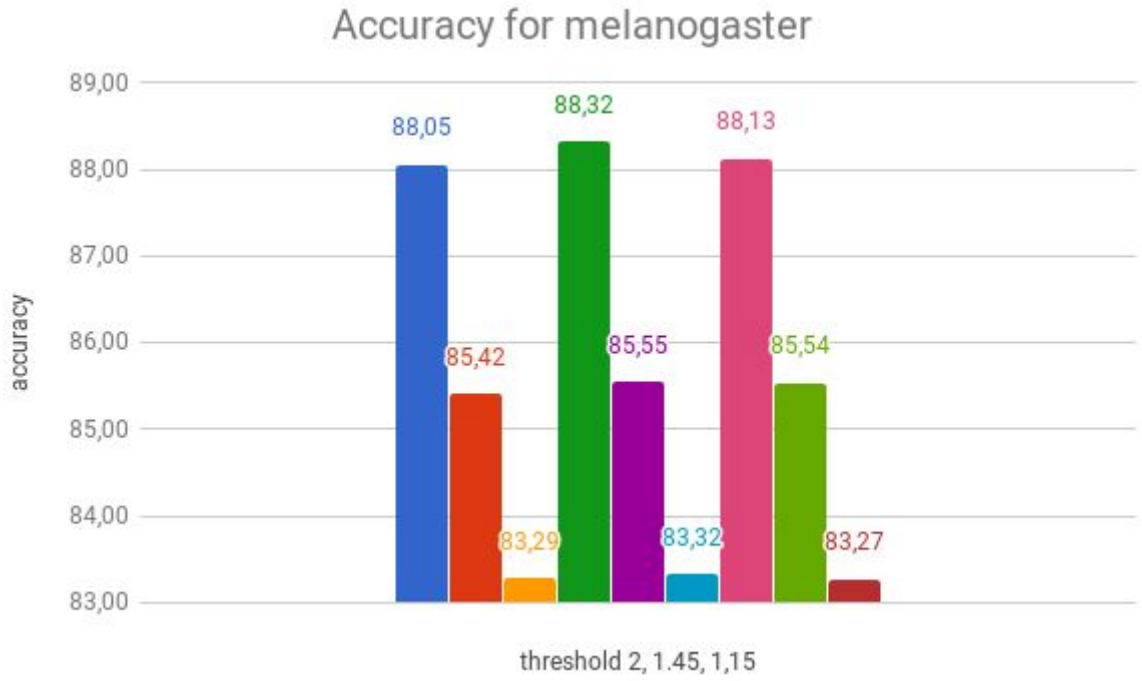

Accuracy for melanogaster
94,00
92,00
90,00
88,00
86,00
accuracy
91,43
88,90
86,92
92,14
89,25
87,22
92,79
89,52
87,43
threshold 2, 1.45, 1,15

Coseno di similarità

k-meri=2

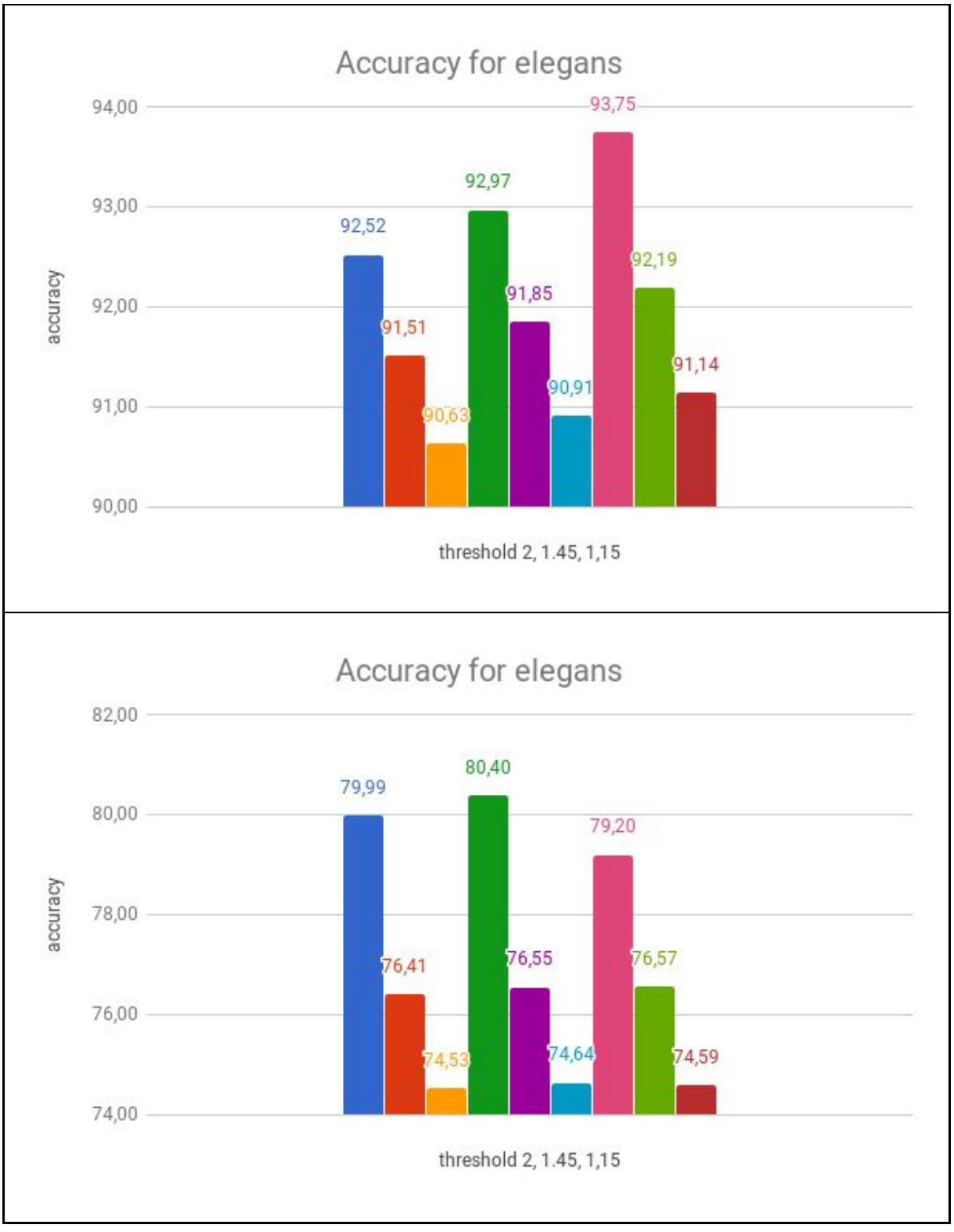


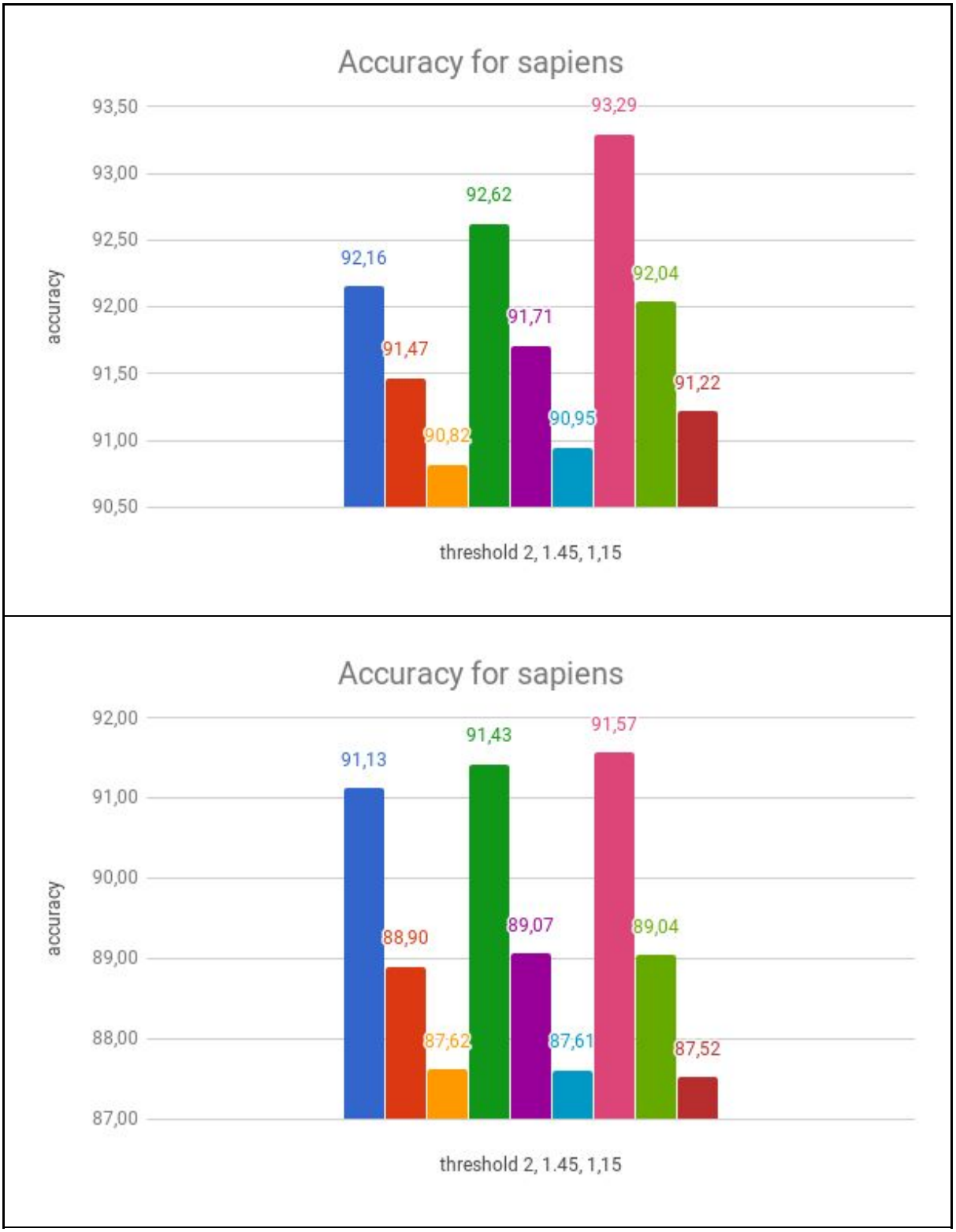


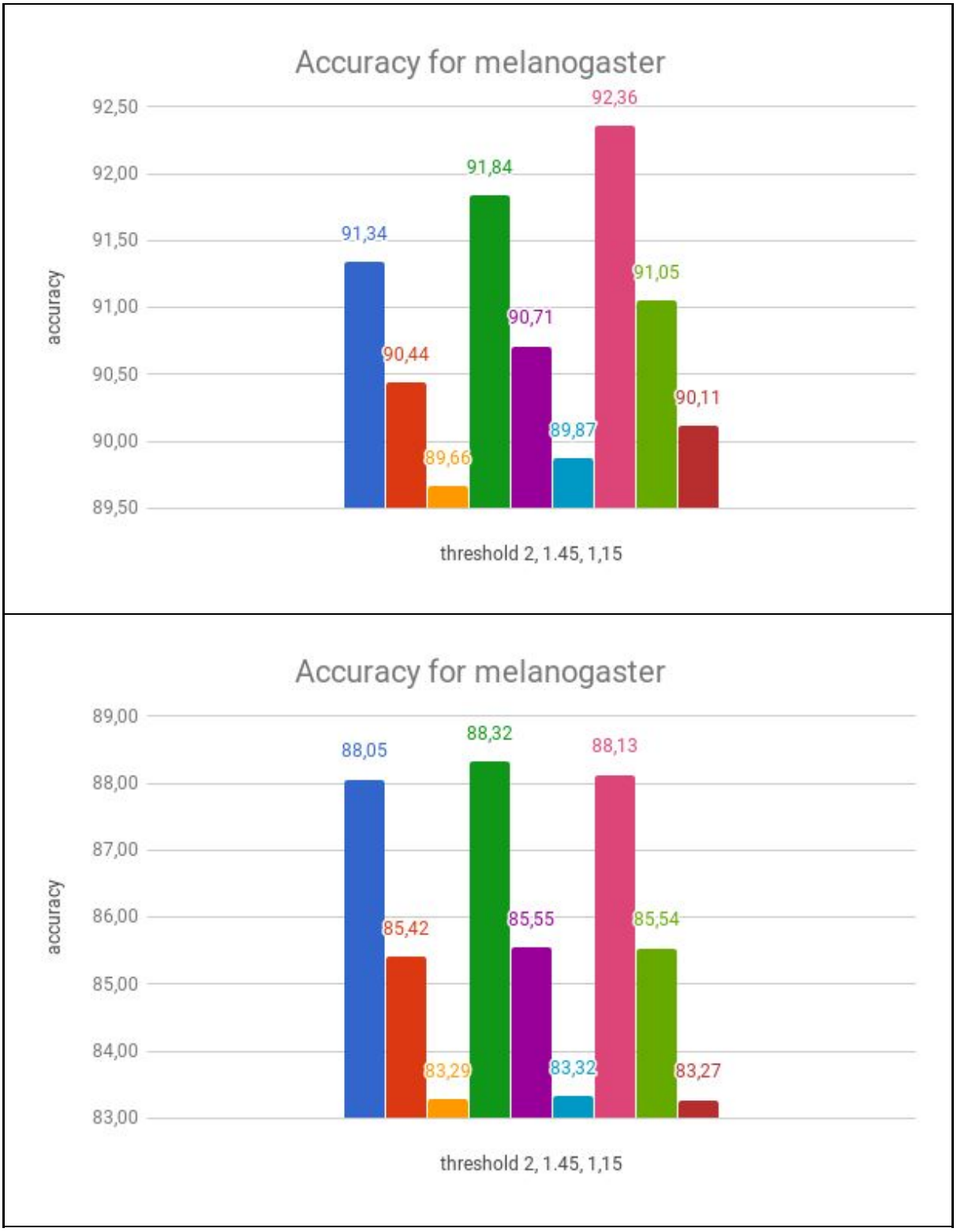

Accuracy for melanogaster
89,00
88,00
87,00
86,00
85,00
84,00
83,00
accuracy
88,05
85,42
83,29
88,32
85,55
83,32
88,13
85,54
83,27
threshold 2, 1.45, 1,15

Comparazione risultati con metrica (Euclidea - coseno)

1°fig Euclidea 2°fig Coseno per k-meri=2
La metrica euclidea mostra un'accuratezza maggiore




Accuracy for sapiens
93,50
93,00
92,50
92,00
91,50
91,00
90,50
accuracy
92,16
91,47
90,82
92,62
91,71
90,95
93,29
92,04
91,22
threshold 2, 1.45, 1,15
Accuracy for sapiens
92,00
91,00
90,00
89,00
88,00
87,00
accuracy
91,13
88,90
87,62
91,43
89,07
87,61
91,57
89,04
87,52
threshold 2, 1.45, 1,15


Accuracy for melanogaster
92,50
92,00
91,50
91,00
90,50
90,00
89,50
accuracy
91,34
90,44
89,66
91,84
90,71
89,87
92,36
91,05
90,11
threshold 2, 1.45, 1,15
Accuracy for melanogaster
89,00
88,00
87,00
86,00
85,00
84,00
83,00
accuracy
88,05
85,42
83,29
88,32
85,55
83,32
88,13
85,54
83,27
threshold 2, 1.45, 1,15

In basso la rappresentazione della curva roc relativa alla macchina addestrata sul dataset nucleosoma_vs_elegans. Come si evince dai risultati, la nostra area d'interesse è sul punto in cui la sensitività è massima tenendo in considerazione che 1-specificità deve rimanere molto bassa (quest'ultima rappresenta la probabilità di rilevare un falso positivo).