LAUREA MAGISTRALE IN INGEGNERIA E SCIENZE INFORMATICHE

CORSO DI TEORIE E TECNICHE DI RICONOSCIMENTO

A.A. 2019/2020

CLASSIFICAZIONE E RICONOSCIMENTO DI MICROARRAY DI DNA

Bertini Diego

Sommario

Motivazione

Il progetto si pone come obiettivo quello di riconoscere le forme tumorali che hanno una componente genetica basandosi su dati provenienti da dei microarray di DNA.

> "Un microarray di DNA (comunemente conosciuto come gene chip, chip a DNA, biochip o matrici ad alta densità) è un insieme di microscopiche sonde di DNA attaccate ad una superficie solida come vetro, plastica, o chip di silicio formanti un array (matrice). Tali array permettono di esaminare simultaneamente la presenza di moltissimi geni all'interno di un campione di DNA" (Wikipedia, Microarray di DNA)

Con questo progetto si vuole mostrare come per alcune tipologie l'analisi e la previsione di patologie tumorali sono possibili attraverso diverse tecniche di riconoscimento e che ogni tecnica ha una sua precisione diversa in base alla quantità e alla tipologia di dato fornito.
La possibilità di fare ricerche su diverse forme di cancro permetterebbe una più facile analisi e prevenzione dei tutti quei tumori subdoli, che si nascondono fino a quando non è troppo tardi per intervenire o anche per dare modo di programmare più spesso esami per coloro che sono riconosciuti predisposti a determinate forme di tumore.

Stato dell'arte

Le tecnologie al momento disponibili per il riconoscimento sono molteplici, ho scelto di focalizzarmi su diverse tecniche per capire quale fosse la più indicata prendendo in considerazione tre diverse tipologie:

- Tecniche di apprendimento supervisionato
- Tecniche di apprendimento non supervisionato
- Tecniche di apprendimento tramite rete neurale

In particolare ho scelto per quanto riguarda le tecniche di apprendimento supervisionato le Support Vector Machines (SVM), una tecnologia che cerca di creare una classificazione tra le diverse classi trovando un iperpiano che separi in modo più preciso possibile le tipologie. Questa tecnica richiede che i dati di training siano espressamente etichettati, ovvero che ci siano un certo numero di campioni dei quali conosco la classe di appartenenza e che verranno utilizzati per addestrare la SVM.
Per quanto riguarda la tecnica di riconoscimento non supervisionata ho pensato di selezionare il metodo del K-means. Diversamente dall'SVM questo metodo non ha bisogno che siano etichettati a priori i dati ma sarà lui a fornire delle etichette in base alla vicinanza di un campione ad un determinato cluster.

Infine ho selezionato una tecnica basata su una rete neurale semplice. Tale tecnologia prende spunto dal funzionamento del cervello, con neuroni e sinapsi, per imparare a distinguere le classi che vengono fornite. Anche questa metodologia necessita di avere dati etichettati ma diversamente dalle precedenti ha bisogno anche di una quantità di dati per il training molto superiore per poter dare dei risultati.
Fatte queste considerazioni ritengo che per poter migliorare ulteriormente la precisione del riconoscimento di queste tecniche sia necessario incrementare il numero di campioni messi a disposizione del training. Inoltre poiché la quantità di features che avevo a disposizione è estremamente grande ho pensato di effettuare una riduzione della dimensionalità delle features utilizzando la tecnica del Principal Component Analisys che per quanto sia efficace è limitata e computazionalmente onerosa, per migliorare la qualità del riconoscimento dopo aver fatto la riduzione bisognerebbe utilizzare un metodo più potente e meno dispendioso di risorse.

Obiettivi

L'obiettivo di questo progetto è ricavare con che grado di accuratezza e precisione è possibile classificare e riconoscere alcune forme tumorali attraverso l'uso e lo studio dei microarray di DNA di una persona e, una volta ottenuti i dati di ogni tecnica, capire quale sia la migliore per il riconoscimento di un determinato tumore.

Metodologia

Dataset

I dataset che ho scelto sono forniti da "*CuMiDa: An Extensively Curated Microarray Database*", un repository di 78 dataset di microarray presi a mano e curati da 30000 studi del Gene Expression Omnibus (GEO) e forniti in formato adatto al machine learning. Nello specifico vengono forniti 5 tipi di file: ARFF, TAB, CSV, GCT e CLS. Di questi tipi ho scelto di utilizzare il csv poiché è quello con cui ho più familiarità.
Ogni dataset è composto da una serie di campioni e per ognuno di questi un microarray dei geni. Il file è formattato come una tabella in cui sulle righe sono disposti i diversi campioni e sulle colonne i diversi geni.
Dei 78 dataset a disposizione ne ho scelti 3 poiché le loro caratteristiche li rendevano abbastanza simili per poterli analizzare con lo stesso programma ma allo stesso tempo abbastanza eterogenei da fornire dei risultati qualitativi della tecnica utilizzata per fare il riconoscimento.

	Tumore al cervello GSE50161	*Tumore al sangue (Leucemia) GSE63270*	*Tumore al fegato GSE14520 U133A*
Campioni	130	281	357
Geni forniti	54676	22284	22278
Numero di Classi	5	7	2

TABELLA 1: INFORMAZIONI SUI DATASET SELEZIONATI

Come è possibile verificare dalla tabella 1 ogni dataset ha un numero diverso di campioni, features e classi. Per quanto riguarda il dataset del tumore al fegato contiene molti più campioni rispetto agli altri due ma relativamente poche features se confrontate con quelle del cervello. Inoltre ha solamente due classi quindi richiede di effettuare una classificazione binaria che è chiaramente più semplice di una classificazione multi-classe, cosa che invece deve essere fatta degli altri due casi. Per quanto riguarda invece il dataset della leucemia è quello mediano tra i 3 che ho selezionato in quanto ha circa lo stesso numero di features del dataset del fegato ma circa lo stesso numero di classi rispetto al dataset del cervello anche se contiene un numero di samples doppio. Infine ho scelto il dataset del cervello per il numero di features che è molto imponente e mi permette quindi di capire se c'è una relazione tra la quantità di features e la precisione e accuratezza del classificatore.
Un'analisi preliminare del dataset mi è inoltre offerta dall'utilizzo della PCA per ridurre a 2 dimensioni l'insieme dei campioni, in modo da permettermi di proiettare i dati su un grafico scatter posizionato in fondo al documento.

Tecniche e Tecnologie

Vista e considerata la quantità di features che i dataset mi forniscono ho come prima cosa pensato di fare una riduzione di dimensionalità cercando di mantenere il maggior numero di informazioni possibili. Per questo task ho selezionato la Principal component analisys – PCA, una tecnica che mira a ridurre la quantità di features cercando di individuare le componenti principali che discriminano le varie classi. In un primo momento ho pensato di implementarne una versione raw in quanto ho ritenuto necessario mantenere una discreta quantità di features ma mi sono subito scontrato con la eccessiva richiesta di risorse che il problema necessitava. Infatti per eseguire la PCA è necessario computare gli autovalori e gli autovettori sulla matrice quadrata delle features che nel mio peggior caso, ovvero il dataset del cervello, risultava essere 54676 x 54676 e richiedeva oltre 22 Gb di memoria di cui io non dispongo.

Sono quindi passato ad una versione sempre usando la versione raw che prevedeva di ridurre la dimensionalità partendo dal numero di campioni invece che delle features. In questo modo il calcolo risultava essere meno oneroso in termini di quantità di memoria ma limitava notevolmente il numero di features finali. Visto il compromesso a cui sono dovuto scendere ho pensato di puntare quindi sull'efficienza passando ad una versione che fa uso di una classe che implementa il PCA fornita dal modulo sklearn. Così il mio numero massimo di features è limitato dalla formula:

$$n\_components \leq \min(n\_features, n\_samples)$$

Questo diventa un problema nel momento in cui vado a dividere il training set dal test set in quanto il numero di features non cambia tra i due set mentre il numero di samples cambia, con 80% del dataset completo che diventa training set e il restante diventa test set. Accettati questi limiti ho deciso di massimizzare comunque il numero di features finali, pur rispettando la formula ottenendo i seguenti risultati.

Database	PCA	Features senza divisione Train/Test	Features con Trainset/testset 80/20
brain	False	54675	54675
	True	130	28
leukemia	False	22283	22283
	True	281	60
liver	False	22277	22277
	True	357	73

Per quanto riguarda invece la classificazione ho scelto tre tecniche:

Support Vector Machines - SVM

Questa tecnologia cerca di trovare un iperpiano che separi in modo più preciso possibile le classi all'interno del dataset. Per fare questo è necessario dividere il dataset in due parti, una parte di training che verrà usata per addestrare la SVM a riconoscere le classi e una parte che verrà usata come campione di test per verificare l'accuratezza della classificazione. L'addestramento del modello avviene passando uno ad uno i campioni all'SVM facendogli sapere determinato campione a che classe si riferisce. Viene infine calcolata sul test set la confusion matrix, l'affidabilità, la precisione e il recall.

K-means

La tecnica del K-means è profondamente diversa rispetto alla SVM in quanto non è richiesto conoscere l'etichetta di un dato per classificarlo. Si tratta infatti di un algoritmo di clusterizzazione che mira a dividere in cluster l'insieme dei campioni basandosi sulla vicinanza al centro di un cluster. In un primo momento i cluster sono decisi in maniera randomica, viene calcolato il centro per ogni cluster e le rispettive distanze dal centro di ogni campione. A questo punto si stabilisce per ogni campione a che cluster appartiene basandoci sulle distanze e si ricomincia con il calcolo dei centri. L'algoritmo viene eseguito fino a raggiungere uno stato di convergenza dei cluster, ovvero in cui i campioni di ogni cluster non vengono più aggiornati da un'iterazione all'altra. Essendo questo un metodo di clusterizzazione non supervisionato i cluster che vengono riconosciuti non hanno alcun riferimento rispetto alle classi che dovrebbero rappresentare. Per creare un matching tra il cluster e la relativa classe di appartenenza ho implementato un sistema di riconoscimento basato numero di etichette all'interno di un cluster. Il cluster che contiene il maggior numero di campioni con una certa lable verrà associato a tale classe. Su questo matching viene quindi calcolata la confusion matrix, l'affidabilità, la precisione e il recall.

Neural Network

Infine come ultimo metodo ho scelto di utilizzare le reti neurali semplici, senza quindi implementare diversi layer nascosti. In particolare la rete che ho creato ha un numero di neuroni input pari al numero di features usate e un numero di neuroni del layer nascosto e di output pari al numero di classi da riconoscere. Per implementare questa architettura ho utilizzato la libreria Pytorch che ho sfruttato per il modello di rete neurale Sequential, in cui ho inserito tre layer: input e output di tipo lineare e layer nascosto di tipo ReLu, ovvero rettificatore lineare. Fatto questo ho definito una funzione di perdita (loss) come la sommatoria quadratica dell'errore. Viene quindi avviata la fase di feed forward con cui alleno la rete usando i campioni di training, quindi viene testata la rete con il test set e calcolato l'errore che verrà propagato (backpropagation) a tutti i nodi per aggiustare la funzione di attivazione degli stessi. Terminata la procedura per diverse volte, dette epoche, viene fatto un ultimo test di predizione della classe del test set e su questo calcolata la confusion matrix, l'affidabilità, la precisione e il recall.

Librerie

Le librerie che vengono usate sono:

- Numpy: per i calcoli sulle matrici ottimizzati
- Sklearn: per le implementazioni di PCA e SVM
- Pytorch: per l'implementazione dei modelli di NN, funzioni di loss e tensori

Risultati

Poiché la divisione dei train set e test set avviene in modo random sono state effettuate 35 esecuzioni della NN per database, ogni esecuzione composta sia dalla versione con PCA che da quella senza. Stesso discorso si applica sia alla versione con K-means che alla versione SVM con 37 esecuzioni, ogni esecuzione dell'SVM è composta da 4 tipologie di kernel e per ogni tipologia di kernel viene utilizzata sia la versione con PCA che quella senza. Per quanto riguarda l'esecuzione della K-means la randomicità viene introdotta non dalla divisione in test set e train set, in quanto non c'è tale differenziazione, ma dall'inizializzazione dei cluster che avviene random. Le confusion matrix non vengono mostrate per una questione organizzativa in quanto avendo un numero così elevato di esecuzioni diverse significherebbe arrivare a saturare con informazioni eccessive il documento. Le tabelle contenenti i risultati sono posizionate in fondo al documento.
Analizzando i dati del database brain riguardanti la SVM possiamo notare che l'accuracy media risulta essere estremamente alta (circa 97%) per quanto riguarda i dati non ridotti di dimensionalità eccetto per la tipologia di kernel sigmoid che dà risultati molto scarsi (33%) nonostante la quantità di features. Utilizzando invece la PCA i risultati sono come ci si aspetta inferiori anche se non sono così pessimi da essere peggio di una scelta casuale (20% poiché ci sono 5 classi). Passando invece al K-means i risultati medi si aggirano intorno al 55% sia con riduzione di dimensionalità che senza. Questo può significare che per quanto riguarda il problema in esame nel database brain la quantità di features usate nell'esecuzione di tale tecnica di riconoscimento non sono particolarmente significative, quanto meno se viene usata la PCA come riduzione. Un risultato del 55% non è comunque male se si considera come punto di riferimento la scelta casuale (20%). Considerando l'ultima tecnica presa in esame, ovvero la Neural network, i risultati sono estremamente scarsi, in particolare se combinata con l'uso della PCA che porta la media ad essere appena inferiore alla scelta casuale. La particolarità dell'utilizzo di questa tecnica è il valore di precision che mentre per i precedenti era molto simile all'accuracy e al recall, i questo caso risulta essere molto alta, contrariamente alla recall che è molto bassa. Questo è un sintomo del fatto che le reti neurali necessitano di avere un dataset molto corposo e sebbene la quantità di features siano estremamente alte, la quantità di campioni è molto scarsa rispetto alle necessità.

Per quanto riguarda invece i risultati del database leukemia analizzando le prestazioni della SVM possiamo notare che come per il precedente database il kernel sigmoid sia il peggiore anche se per tale kernel i risultati post PCA sono notevolmente superiori a quelli senza PCA, anche se comunque lontani dalla accuratezza degli altri kernel. Anche per quanto riguarda il K-means l'esito è similare al precedente database nonostante l'aumento delle classi richieste. Inoltre come per brain, la differenza tra l'uso o il non uso del PCA è minimo e quindi porta alle precedenti conclusioni. Infine considerando la Neural network abbiamo valori similari al database brain, ovvero un'accuracy assolutamente non soddisfacente perché molto vicina alla scelta casuale. Come per il precedente dataset la precision risulta essere molto alta rispetto ad accuracy e recall.
Analizzando ora il database liver ci sono risultati in parte simili ai precedenti ed in parti diversi. Innanzitutto le differenze sono che l'accuracy del kernel sigmoid, i più marginalmente nel kernel rbf, è molto alta rispetto ai precedenti per quanto riguarda il riconoscimento con PCA mentre è di poco superiore alla scelta casuale (50%) se non viene ridotta la dimensionalità. Ciò dipende dal fatto che le features che vengono rimosse sono fuorvianti per il riconoscimento della classe di appartenenza ed eliminarle non risulta controproducente come invece avveniva per i precedenti dataset. Le similarità invece si trovano nei valori di accuracy degli altri kernel che risultano essere molto alti sia in caso di mancato uso della PCA che se viene utilizzata, anche se si perde comunque un po' di informazione e accuratezza. Per quanto riguarda l'analisi dei risultati del K-means otteniamo un risultato stazionario del 94% di accuracy, precision e recall sia senza PCA che con. Non è un risultato inaspettato se si guarda al grafico 2D della PCA in quanto abbiamo due classi che sono molto separate con pochissime eccezioni nel mezzo. L'ultima analisi che rimane riguarda la Neural network che, sebbene i risultati siano attorno al 60% non sono soddisfacenti in quanto non lontani dal caso. Questo dimostra ancora una volta che se viene fornito un dataset troppo piccolo le prestazioni delle reti neurali non dipendono dal numero di classi da riconoscere ma principalmente dalla quantità di campioni che vengono usati per allenare la rete.

Conclusioni

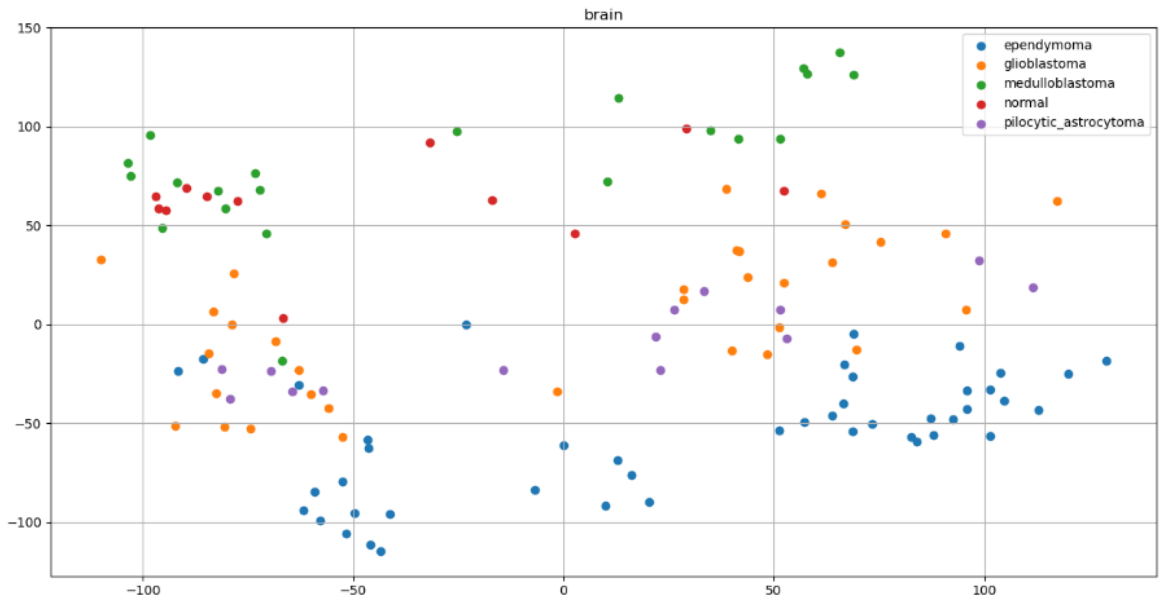
Per quanto riguarda il dataset brain possiamo affermare che la tecnica SVM sia indubbiamente la più adatta poiché l'accuracy, la precision e la recall sono senza dubbio le migliori mentre la meno adatta sia la NN in quanto risulta essere molto svantaggiosa sotto ogni punto di vista, sia dei risultati che della complessità computazionale.
Nel caso del dataset leukemia la scelta della tecnica più adatta ricade come per il precedente sulla SVM, avendo una accuratezza molto alta. La similarità dei risultati tra il dataset brain e leukemia è da attribuire alla caratteristica del problema di essere un riconoscimento multi-class. Tale caratteristica risulta essere predominante rispetto alla differenza di quantità di features e samples dei due database in esame.

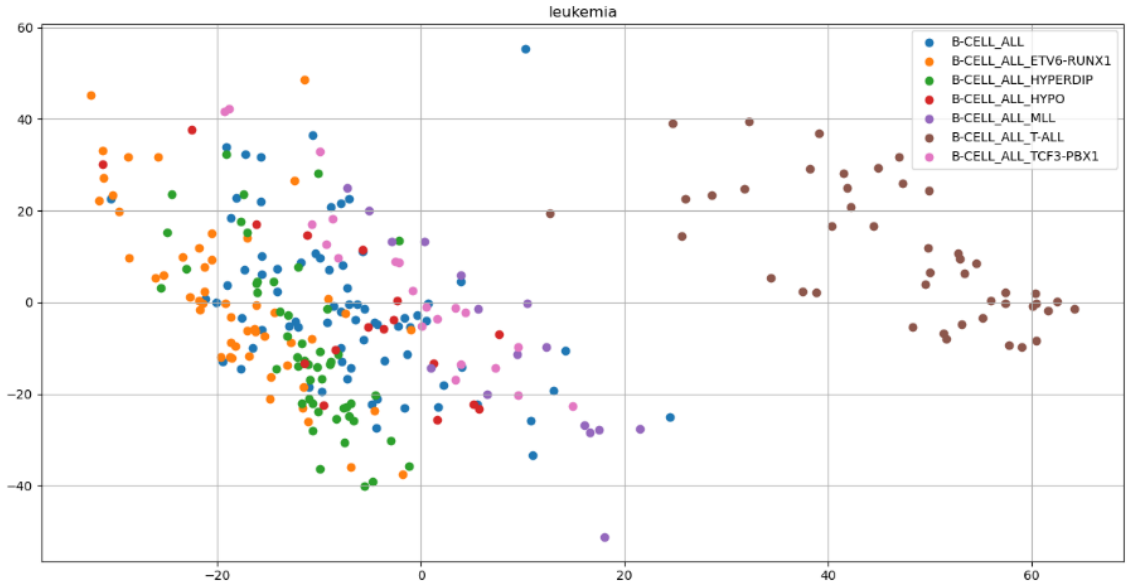
Infine per il dataset liver la scelta del metodo migliore è piuttosto combattuta in quanto sia SVM che K-means hanno risultati importanti. Nel caso in cui un eventuale incremento del dataset non va a confondere la linea di separazione delle due classi allora la scelta potrebbe ricadere su K-means ricordando però che il riconoscimento vero e proprio delle classi avviene per maggioranza di campioni nel cluster quindi le prestazioni potrebbero drasticamente peggiorare se questo metodo dovesse risultare inefficace. Per avere invece un riconoscimento più affidabile anche se meno intuitivo la scelta ricadrebbe sulla SVM con kernel linear i quanto garantisce un valore di accuracy, recall e precision molto vicino al 100%.

Sitografia

- Informazioni Microarray di DNA: https://it.wikipedia.org/wiki/Microarray_di_DNA
- Dataset CuMiDa: http://sbcb.inf.ufrgs.br/cumida

Scatter PCA




leukemia
B-CELL_ALL
B-CELL_ALL_ETV6-RUNX1
B-CELL_ALL_HYPERDIP
B-CELL_ALL_HYPO
B-CELL_ALL_MLL
B-CELL_ALL_T-ALL
B-CELL_ALL_TCF3-PBX1
60
40
20
0
−20
−40
−20
0
20
40
60

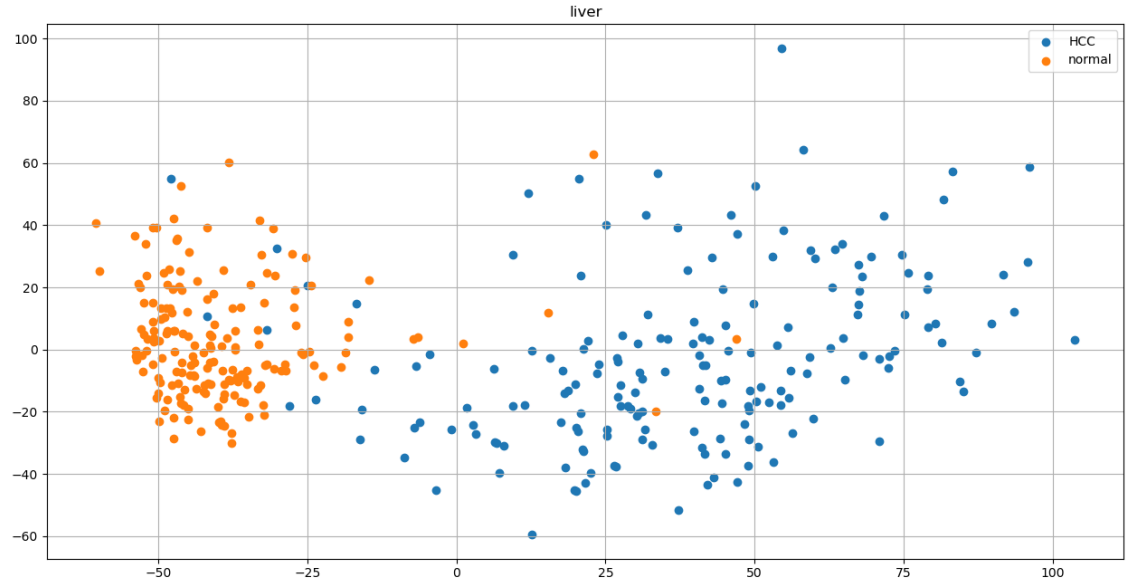

liver
HCC
normal
100
80
60
40
20
0
−20
−40
−60
−50
−25
0
25
50
75
100

Tabelle Risultati

SVM

Database	Kernel	PCA		accuracy	precision	recall
brain	linear	False	max	1	1	1
			min	0,88	0,88	0,89
			mean	0,97	0,98	0,98
		True	max	0,64	0,63	0,6
			min	0,03	0,02	0,02
			mean	0,32	0,27	0,26
	poly	False	max	1	1	1
			min	0,87	0,92	0,87
			mean	0,97	0,97	0,97
		True	max	0,41	0,65	0,37
			min	0,2	0,14	0,15
			mean	0,29	0,44	0,22
	rbf	False	max	1	1	1
			min	0,88	0,88	0,83
			mean	0,96	0,96	0,96
		True	max	0,76	0,88	0,73
			min	0	0,04	0
			mean	0,31	0,40	0,27
	sigmoid	False	max	0,61	0,91	0,7
			min	0,12	0,46	0,2
			mean	0,33	0,62	0,37
		True	max	0,64	0,66	0,61
			min	0,03	0,03	0,04
			mean	0,34	0,30	0,29

Database	Kernel	PCA		accuracy	precision	recall
leukemia	linear	False	max	0,9	0,94	0,9
			min	0,65	0,75	0,67
			mean	0,77	0,84	0,79
		True	max	0,45	0,53	0,43
			min	0,17	0,15	0,12
			mean	0,30	0,28	0,24
	poly	False	max	0,91	0,94	0,93
			min	0,66	0,76	0,69
			mean	0,77	0,84	0,80
		True	max	0,24	0,53	0,2
			min	0,16	0,22	0,13
			mean	0,19	0,35	0,17
	rbf	False	max	0,88	0,93	0,84
			min	0,49	0,68	0,47
			mean	0,67	0,80	0,70
		True	max	0,62	0,8	0,62
			min	0,22	0,31	0,16
			mean	0,40	0,57	0,32
	sigmoid	False	max	0,27	0,88	0,37
			min	0,08	0,18	0,14
			mean	0,17	0,39	0,21
		True	max	0,57	0,52	0,47
			min	0,17	0,13	0,13
			mean	0,32	0,29	0,25

Database	Kernel	PCA		accuracy	precision	recall
liver	linear	False	max	1	1	1
			min	0,89	0,9	0,89
			mean	0,97	0,97	0,97
		True	max	0,95	0,95	0,95
			min	0,68	0,69	0,68
			mean	0,83	0,84	0,83
	poly	False	max	1	1	1
			min	0,89	0,9	0,89
			mean	0,96	0,97	0,96
		True	max	0,86	0,89	0,87
			min	0,5	0,5	0,5
			mean	0,58	0,62	0,60
	rbf	False	max	0,97	0,97	0,97
			min	0,66	0,68	0,68
			mean	0,84	0,85	0,85
		True	max	0,99	0,99	0,99
			min	0,77	0,78	0,77
			mean	0,91	0,91	0,91
	sigmoid	False	max	0,89	0,89	0,88
			min	0,44	0,36	0,44
			mean	0,63	0,66	0,63
		True	max	0,97	0,97	0,97
			min	0,68	0,69	0,68
			mean	0,85	0,86	0,85

K-means

Database	PCA		accuracy	precision	recall
brain	False	max	0,68	0,76	0,72
		min	0,38	0,35	0,28
		mean	0,55	0,58	0,55
	True	max	0,68	0,76	0,72
		min	0,36	0,34	0,28
		mean	0,54	0,56	0,53

Database	PCA		accuracy	precision	recall
leukemia	False	max	0,72	0,72	0,7
		min	0,23	0,27	0,17
		mean	0,53	0,50	0,49
	True	max	0,74	0,76	0,73
		min	0,35	0,32	0,27
		mean	0,55	0,52	0,50

Database	PCA		accuracy	precision	recall
liver	False	max	0,94	0,94	0,94
		min	0,94	0,94	0,94
		mean	0,94	0,94	0,94
	True	max	0,94	0,94	0,94
		min	0,94	0,94	0,94
		mean	0,94	0,94	0,94

Neural Network

Database	PCA		accuracy	precision	recall
brain	False	max	0,75	0,87	0,57
		min	0,11	0,62	0,2
		mean	0,26	0,80	0,25
	True	max	0,32	0,51	0,29
		min	0	0,02	0
		mean	0,19	0,24	0,17

Database	PCA		accuracy	precision	recall
leukemia	False	max	0,7	0,9	0,59
		min	0,07	0,57	0,13
		mean	0,34	0,80	0,29
	True	max	0,35	0,47	0,4
		min	0,1	0,08	0,07
		mean	0,20	0,24	0,19

Database	PCA		accuracy	precision	recall
liver	False	max	0,99	0,99	0,99
		min	0,49	0,75	0,5
		mean	0,60	0,80	0,60
	True	max	0,93	0,93	0,93
		min	0,45	0,43	0,46
		mean	0,64	0,71	0,64