



DIPARTIMENTO
DI INFORMATICA

SISTEMA BASATO SU CONOSCENZA PER LA CLASSIFICAZIONE DELLA DEMENTIA SENILE



Dementia Cognitive Test

Corso di Studi: Ingegneria della Conoscenza

Docente: Fanizzi Nicola

Studente: Zingaro Felice

Matricola: 660972

Sommario

Introduzione.....	3
Informazioni sul dataset utilizzato	3
Mini-Mental State Examination (MMSE).....	3
CDR	4
Estimated total intracranial volume (eTIV)	4
Normalized whole-brain volume (nWBV).....	4
Atlas Scaling Factor (ASF).....	4
Elaborazione dei dati.....	4
Analisi dei dati	6
Apprendimento supervisionato dei classificatori	7
Decision Tree	8
K-Nearest Neighbors	9
Random Forest	10
Logistic Regression	11
Multi Layer Perceptron.....	12
Apprendimento non supervisionato	13
Clustering K-Means	13
Conclusioni.....	14
Interfaccia a riga di comando (CLI).....	15
Menù.....	15
Learning	15
Optimize Params.....	16
Run Test	16
Sviluppi Futuri	17

Introduzione

Per demenza si intende un termine generale che indica un declino delle capacità mentali molto gravi da interferire con la vita quotidiana. La perdita di memoria ne è un esempio.

La demenza non è una malattia specifica, ma un termine generale che descrive un gruppo di sintomi associata ad un declino della memoria o di altre capacità mentali abbastanza grave da ridurre la capacità di una persona di svolgere le attività quotidiane.

Informazioni sul dataset utilizzato

Il dataset utilizzato per l'addestramento del sistema è tratto dal sito [www.kaggle.com](https://www.kaggle.com/code/obrienmitch94/alzheimer-s-analysis/report) ed è visualizzabile tramite il seguente link

<https://www.kaggle.com/code/obrienmitch94/alzheimer-s-analysis/report>

Questo dataset si basa su una raccolta longitudinale di 150 soggetti di età compresa tra 60 e 96 anni.

Ciascun soggetto del database è stato sottoposto a risonanza magnetica pesata in T1 in due o più visite, a distanza di almeno un anno, per un totale di 373 scansioni di imaging.

Sono presenti 9 feature binarie che rappresentano dati vari, e 5 feature di categoria.

<i>Feature</i>	<i>Description</i>
<i>ID</i>	Identificativo
<i>Group</i>	Stato della demenza
<i>Visit</i>	Numero della visita
<i>M/F</i>	Sesso del paziente
<i>Hand</i>	Mano predominante
<i>Age</i>	Età del paziente
<i>Educ</i>	Anni di educazione
<i>SES</i>	Stato economico
<i>MMSE</i>	Esame ridotto dello stato mentale
<i>CDR</i>	Valutazione della demenza clinica
<i>eTIV</i>	Volume intracranico totale
<i>nWBV</i>	Volume normalizzato dell'intero cervello
<i>ASF</i>	Atlas Scaling Factor

Mini-Mental State Examination (MMSE)

Il Mini-Mental State Examination (MMSE) o test di Folstein è un questionario di 30 punti ampiamente utilizzato in ambito clinico e di ricerca per misurare il deterioramento cognitivo effettuando anche uno screening della demenza.

Utilizzato inoltre per stimare la gravità e la progressione del deterioramento cognitivo di un individuo nel corso del tempo (e quindi un metodo efficace per documentare la risposta di un individuo al trattamento.)

Alla fine del questionario, è attribuito uno punteggio: qualsiasi punteggio superiore o uguale a 24 punti (su 30) indica una cognizione normale. Al di sotto, i punteggi possono indicare un deterioramento cognitivo grave (≤ 9 punti), moderato

(10-18 punti) o lieve (19-23 punti). Il punteggio grezzo può anche dover essere corretto in base al livello di istruzione e all'età. In altre parole, un punteggio massimo di 30 punti non può mai escludere la demenza. I punteggi da bassi a molto bassi sono strettamente correlati alla presenza di demenza, sebbene anche altri disturbi mentali possano portare a risultati anomali nei test MMSE. Anche la presenza di problemi puramente fisici può interferire con l'interpretazione se non adeguatamente segnalata; per esempio, un paziente può essere fisicamente incapace di ascoltare o leggere correttamente le istruzioni, oppure può avere un deficit motorio che influisce sulle capacità di scrittura e disegno.

CDR

La CDR è una scala a cinque punti utilizzata per caratterizzare sei domini di prestazioni cognitive e funzionali applicabili alla malattia di Alzheimer e alle demenze correlate: Memoria, Orientamento, Giudizio e Risoluzione dei problemi, Affari comunitari, Casa e Hobby e Cura della persona. Le informazioni necessarie per effettuare ciascuna valutazione sono ottenute attraverso un'intervista semi-strutturata col paziente e con un informatore affidabile o fonte correlabile (per esempio, un familiare), denominata Protocollo di valutazione CDR.

I valori che può assumere sono:

0 → Non demente

0,5 → Inizio di demenza

1 → Demenza lieve

2 → Demenza moderata

3 → Demenza grave

Estimated total intracranial volume (eTIV)

La misura dell'ICV, talvolta indicata come volume intracranico totale (TIV), si riferisce al volume stimato della cavità cranica come delineato dalla materia della dura sopratentoriale o dal contorno cerebrale quando la dura non è chiaramente individuabile.

La costanza di questo parametro durante l'invecchiamento lo rende uno strumento affidabile per l'analisi della variazione della concavità cranica tra i soggetti in quanto gli studi si basano sulle morfologie del cervello. L'ICV, insieme all'età e al sesso, viene utilizzato come covariata per studiare le analisi di regressione nello studio dei disturbi cerebrali neurodegenerativi progressivi, quali la malattia di Alzheimer, l'invecchiamento e il deterioramento cognitivo.

Normalized whole-brain volume (nWBV)

nWBV è una misura espressa come la percentuale di tutti i voxel ¹ di un atlas ² che sono etichettati come materia grigia o bianca dal processo di segmentazione automatica dei tessuti.

Atlas Scaling Factor (ASF)

Si tratta di un fattore di scala calcolato per la trasformazione del cervello e del cranio nello spazio nativo al target dell'atlante.

Elaborazione dei dati

La feature Group contiene solo valori specifici come "Nondemented, Demented, Converted". Quest'ultimo valore rappresenta tutti quei pazienti che inizialmente non presentavano alcun sintomo di demenza senile, ma successivamente si sono rilevati anch'essi affetti da questo disturbo; per essi cui verranno rappresentati come "Demented".

Quindi, in base alla condizione attuale, l'insieme dei dati è ora separato in due classi, dove "Nondemented" verrà etichettato come 0 e "Demented" con 1.

La stessa operazione è stata fatta anche per le feature M/F, rispettivamente 0 e 1, e per HAND, che conteneva solo valori R/L, ovvero 0,1.

Per eseguire il progetto abbiamo sfruttato i seguenti elementi:

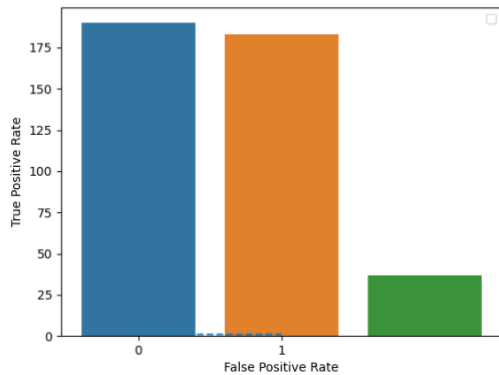
- PyCharm

¹ Un voxel (volumetric picture element) è un'unità di misura del volume.

² Un atlas cerebrale è composto da sezioni seriali lungo diversi piani anatomici del cervello umano o animale, sano o malato, in fase di sviluppo o adulto, in cui a ogni struttura cerebrale rilevante viene assegnato un numero di coordinate per definirne il contorno o il volume.

- Python
- Librerie quali:
 - Numpy: per l'utilizzo delle relative funzioni matematiche
 - Pandas: per la gestione dei dataframe
 - Matplotlib, Seaborn e Plotly: per la generazione dei grafici e delle tabelle
 - Scikit-learn: per la fase di classificazione e le metriche di valutazione

Analisi dei dati



Una volta determinato il dataset da utilizzare, è stato realizzato un istogramma contenente le distribuzioni dei casi rispetto al dataset nella sua interezza al fine di determinare se i dati presenti fossero equamente distribuiti.

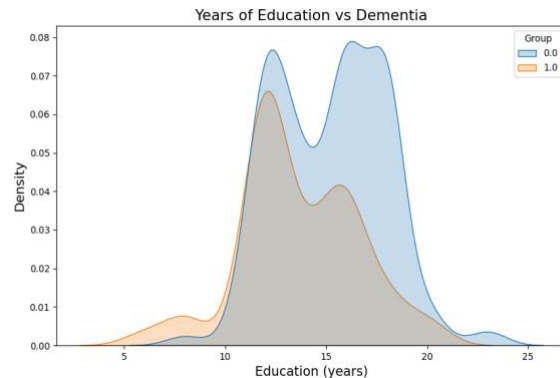
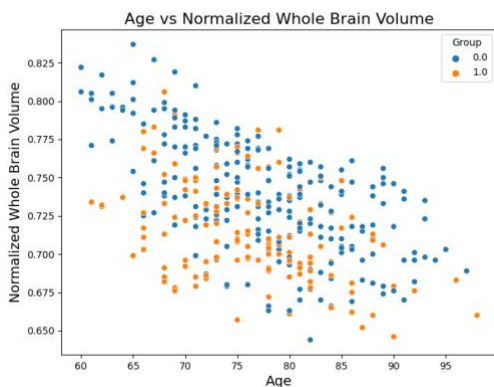
Il risultato è che sono presenti in modo equivalente sia pazienti affetti da demenza senile, sia non.

Inoltre, sono riportati anche i casi di paziente che sono stati successivamente dichiarati affetti da demenza senile (rappresentati in verde).

Quest'ultimi, per semplicità, sono stati convertiti come soggetti affetti da demenza.

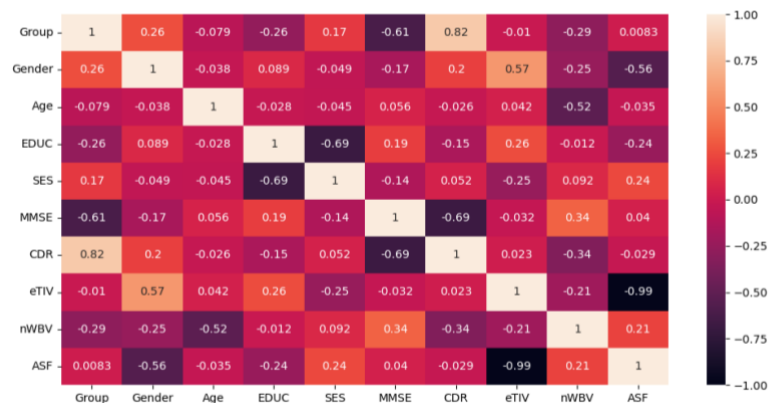
Una volta presa visione del fatto che la distribuzione dei dati è equa in tutto il dataset, si è andati alla ricerca di fattori che andassero a caratterizzare il paziente affetto da Alzheimer.

Sulla base dei features del nostro dataset, le caratteristiche che possono far pensare ad una possibile causa della demenza senile sono l'età, gli anni di studio o il suo lavoro; per cui gli andremo a studiare su un grafico dedicato.



Dal risultato ottenuto, possiamo notare che questi due fattori non rappresentino una caratteristica importante per la demenza senile.

Dato che non è stato possibile trarre delle affinità della demenza con queste caratteristiche, è stato generata una heatmap che ci permetterà di visualizzare quali sono i sintomi più comuni correlati tra di loro.



Dalla heatmap si nota, dunque, che ci sono sintomi mediamente o altamente correlati tra di loro.

Se ne deduce quindi che per gli esempi del dataset che avranno valore 1 in corrispondenza di feature mediamente o altamente correlate tra di loro, la classificazione sarà più attendibile e veritiera.

Dato che l'obiettivo del nostro lavoro è quello di realizzare un sistema che possa essere di supporto ad un medico (e non ad uso e consumo di un privato), abbiamo concentrato i nostri sforzi nell'addestrare una serie di classificatori e nel determinarne i migliori iper-parametri al fine di poter fornire al medico una serie di prognosi piuttosto che una sola.

Apprendimento supervisionato dei classificatori

Dal momento che il nostro dataset per ogni paziente presenta un numero quasi equo di casistiche "demente" / "non demente", si è optato per l'addestramento dei classificatori solo sul sesso, età, educazione, SES, MMSE, eTIV, nWBV, ASF, tralasciando features quali la mano predominante (i pazienti sono tutti destrorsi) e gli identificativi.

Il dataset è stato quindi suddiviso in quattro insiemi: due per la fase di training e due per la fase di testing con una proporzione rispettivamente del 75% e del 25%.

La scelta dei classificatori è ricaduta su:

- Decision Tree
- K-Nearest Neighbors
- Logistic Regression
- Random Forest
- Neural Network

La ricerca degli iperparametri è stata effettuata tramite la Grid Search, andando a definire per ogni classificatore una lista di quelli che sono gli attributi e per ogni attributo è stato specificato un insieme di valori assumibili.

Ciò che fa la Grid Search è testare il classificatore con ogni possibile combinazione degli attributi specificatigli.

Per ogni combinazione effettua una k-fold cross validation (con un valore k settato arbitrariamente a cinque in quanto è la profondità massima che si può raggiungere) e calcola lo score di accuratezza relativo all'iterazione corrente.

Una volta che sono state testate tutte le combinazioni viene restituita la combinazione con l'accuracy score maggiore.

La scelta per la valutazione dei classificatori è ricaduta su:

Matrice di confusione e *0-1 Loss* per la valutazione delle classificazioni rispetto alle ground truth

Classification Report per visualizzare precision, recall, f1-score, accuracy, macro average e weighted average per ogni classificatore

Cross Validation Score per calcolare lo score della cross-validation

Learning Curve per valutare l'accuratezza rispetto all'aumentare del numero di esempi sottoposti ai classificatori

Decision Tree

Il funzionamento degli alberi di decisione prevede che il valore di una feature obiettivo venga classificato sulla base di una serie di regole di decisione basate sui dati di input a disposizione.

Nello specifico ogni nodo interno dell'albero indica una condizione ed i valori derivati dagli esempi di input costituiscono dei

sottoalberi. Le foglie dell'albero invece contengono il valore della feature obiettivo.

Per quanto riguarda la nostra implementazione di questo modello di classificazione, a seguito degli insoddisfacenti risultati iniziali, abbiamo effettuato subito una Grid Search (operazione che abbiamo effettuato per prima a partire da questo classificatore) dalla quale è risultato che il miglior criterio tra "entropy" e "gini" è proprio quest'ultimo.

Il criterio "gini" permette di minimizzare la probabilità di classificazioni errate e la sua formula è:

$$Gini = 1 - \sum_x P(x)^2$$

Per quanto riguarda la profondità massima dell'albero, la Grid Search ha suggerito una profondità di 14 nodi poiché valori più alti avrebbero sicuramente decretato classificazioni più accurate in fase di training ma avrebbero portato ad un problema di overfitting nella fase di test.

	WITHOUT FINE OPTIMIZATION	WITH FINE-OPTIMIZATION
TRAIN SCORE	0,79	1
TEST SCORE	0,73	0,86
DEVIAZIONE STANDARD	0,07	0,08
VARIANZA	0,004	0,007
0-1 LOSS	0,26	0,13
ACCURACY	0,74	0,86

Classification report con ottimizzazione dei parametri:

	PRECISION	RECALL	F1 SCORE	SUPPORT
NONDEMENTED	0.86	0,86	0.86	35
DEMENTED	0.88	0.88	0.88	40
ACCURACY			0.77	94
MACRO AVG	0.77	0.77	0.77	94
WEIGHTED AVG	0.77	0.77	0.77	94

K-Nearest Neighbors

Il funzionamento dell'algoritmo relativo al classificatore K-NN si basa sulla semplice memorizzazione degli esempi del dataset (comprendendo la/le feature obiettivo), senza che venga appreso un modello.

La classificazione avviene confrontando il nuovo esempio con un insieme formato per l'appunto da k vicini che voteranno per decretare la classe di appartenenza del nuovo esempio.

La votazione può avvenire come calcolo della moda, della media o a seguito dell'interpolazione dei k vicini.

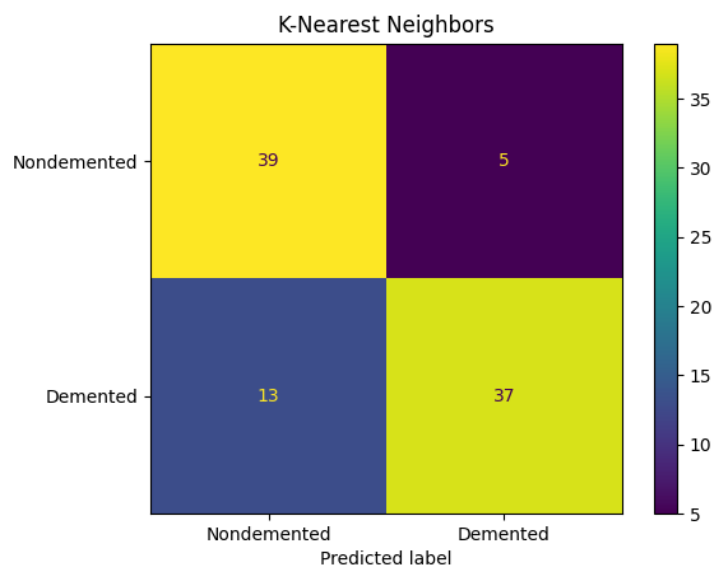
I risultati di fatto sono stati:

	WITHOUT FINE OPTIMIZATION	WITH FINE-OPTIMIZATION
TRAIN SCORE	0.71	1
TEST SCORE	0.78	0.80
DEVIAZIONE STANDARD	0.30	0.04
VARIANZA	0.0009	0.001
0-1 LOSS	0.21	0.19
ACCURACY	0.79	0.81

Classification report con ottimizzazione dei parametri:

	PRECISION	RECALL	F1 SCORE	SUPPORT
NONDEMENTED	0.75	0.89	0.81	44
DEMENTED	0.88	0.84	0.80	50
ACCURACY			0.81	94
MACRO AVG	0.82	0.81	0.81	94
WEIGHTED AVG	0.82	0.81	0.81	94

Per avere una rassicurazione circa il corretto apprendimento del classificatore, è stata creata una matrice di confusione, la quale ci ha fornito una rappresentazione grafica dell'accuratezza del classificatore:



Random Forest

L'algoritmo dell'estimatore Random Forest è basato sull'addestramento di un numero N di Decision Tree, ognuno dei quali effettua una classificazione per ogni esempio.

Quando tutti gli alberi (o più precisamente tutta la foresta) hanno classificato l'esempio, si effettua una conta di qual è stata la classe maggiormente stimata e la si assume come predizione della foresta. Questo per l'appunto viene effettuato per tutti gli esempi.

Dati gli ottimi risultati ottenuti con la Grid Search abbiamo ripetuto la ricerca degli iper-parametri anche per questo estimatore ed è risultato che il numero di alberi più adatto per la foresta è pari a 90, per ogni albero che compone la foresta si evince che il miglior criterio sia "entropy" (diversamente dagli alberi di decisione dove è risultato più valido il criterio "gini").

La formula relativa all'entropia è la seguente:

$$Entropy = \sum_x -P(x) * \log_2 P(x)$$

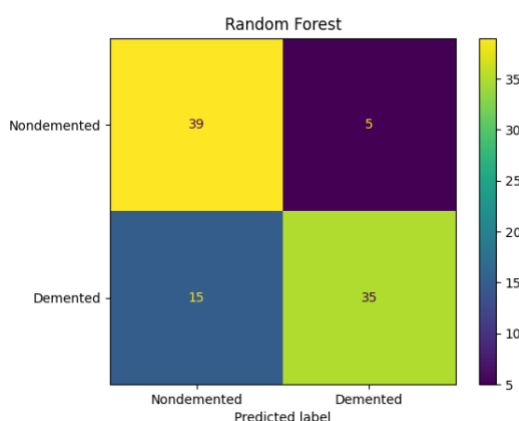
I parametri migliori indicati dal Grid Search sono: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': 9, 'max_features': 'sqrt', 'n_estimators': 90}

	WITHOUT FINE OPTIMIZATION	WITH FINE-OPTIMIZATION
TRAIN SCORE	0.87	1
TEST SCORE	0.84	0.86
DEVIAZIONE STANDARD	0.10	0.08
VARIANZA	0.01	0.007
0-1 LOSS	0.18	0.13
ACCURACY	0.81	0.86

Classification report con ottimizzazione dei parametri:

	PRECISION	RECALL	F1 SCORE	SUPPORT
NONDEMENTED	0.80	0.93	0.86	44
DEMENTED	0.93	0.80	0.86	50
ACCURACY			0.86	94
MACRO AVG	0.87	0.87	0.86	94
WEIGHTED AVG	0.87	0.86	0.86	94

Per avere una rassicurazione circa il corretto apprendimento del classificatore, è stata creata una matrice di confusione, la quale ci ha fornito una rappresentazione grafica dell'accuratezza del classificatore:



Logistic Regression

Il modello di classificazione della regressione logistica si basa su dei pesi di una funzione lineare appiattita dalle sigmoidee, minimizzando un errore su E.

È importante specificare che parte come un modello di regressione, ma successivamente viene appiattito con la log loss.

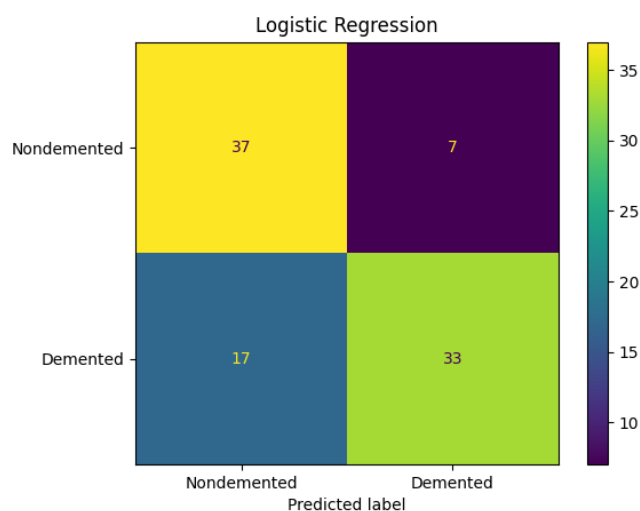
I parametri migliori indicati dal Grid Search sono: {'bootstrap': False, 'criterion': 'entropy', 'max_depth': 9, 'max_features': 'sqrt', 'n_estimators': 90}

	WITHOUT FINE OPTIMIZATION	WITH FINE-OPTIMIZATION
TRAIN SCORE	0.82	0.82
TEST SCORE	0.81	0.74
DEVIAZIONE STANDARD	0.10	0.09
VARIANZA	0.01	0.009
0-1 LOSS	0.18	0.25
ACCURACY	0.81	0.75

Classification report con ottimizzazione dei parametri:

	PRECISION	RECALL	F1 SCORE	SUPPORT
NONDEMENTED	0.69	0.84	0.76	44
DEMENTED	0.82	0.66	0.73	50
ACCURACY			0.74	94
MACRO AVG	0.76	0.75	0.74	94
WEIGHTED AVG	0.76	0.76	0.74	94

Per avere una rassicurazione circa il corretto apprendimento del classificatore, è stata creata una matrice di confusione, la quale ci ha fornito una rappresentazione grafica dell'accuratezza del classificatore:



Multi Layer Perceptron

Una Rete Neurale è un modello di apprendimento che si basa sul funzionamento dei neuroni cerebrali biologici, per il nostro problema abbiamo utilizzato una Rete Neurale di tipo **feed-forward** che si basa su una gerarchia di funzioni lineari intervallate da funzioni di attivazione. In genere prende in input una serie di feature e le sottopone agli strati nascosti (detti anche feature non osservate) che le mappano secondo una funzione di attivazione e restituiscono in output uno o più feature obiettivo. Formalmente è composta da tre strati o layer:

Un layer di input (nel nostro caso tutte le feature che rappresentano i sintomi).

Un layer lineare completo, il cui modello matematico è

$$y = \sum_i w_i * x_i$$

Dove y è la classe da predire, x è il numero di feature di input e w sono i pesi da assegnare ad ogni feature di input.

Una funzione di attivazione f .

Per ottenere una classificazione accurata si fa uso della **back-propagation** che ricalcola ed aggiorna i pesi w .

Abbiamo optato per l'uso di una Rete Neurale poiché questa tipologia di modello è flessibile e si comporta molto bene nel caso si disponga di molti dati.

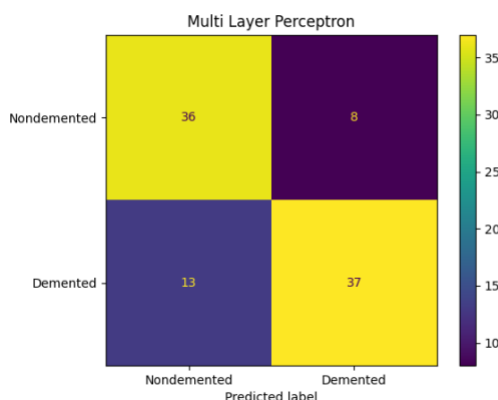
Dato l'elevato costo computazionale che una rete neurale comporta, ci siamo limitati ad effettuare dei test per la ricerca della migliore funzione di attivazione per la nostra rete.

	WITHOUT FINE OPTIMIZATION	WITH FINE-OPTIMIZATION
TRAIN SCORE	0.87	0.89
TEST SCORE	0.76	0.77
DEVIAZIONE STANDARD	0.10	0.015
VARIANZA	0.01	0.0002
0-1 LOSS	0.18	0.22
ACCURACY	0.81	0.77

Classification report con ottimizzazione dei parametri:

	PRECISION	RECALL	F1 SCORE	SUPPORT
NONDEMENTED	0.73	0.82	0.77	44
DEMENTED	0.82	0.74	0.78	50
ACCURACY			0.74	94
MACRO AVG	0.78	0.78	0.78	94
WEIGHTED AVG	0.78	0.78	0.78	94

Per avere una rassicurazione circa il corretto apprendimento del classificatore, è stata creata una matrice di confusione, la quale ci ha fornito una rappresentazione grafica dell'accuratezza del classificatore:



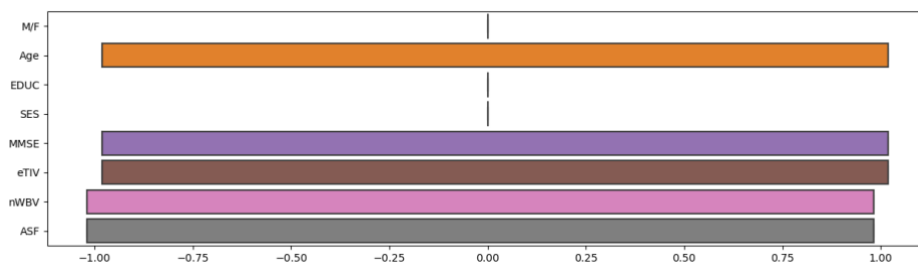
Apprendimento non supervisionato

Per apprendimento non supervisionato si intende un apprendimento in cui non ci sono feature-obiettivo negli esempi di training.

Clustering K-Means

Questo algoritmo separa i dati in cluster distinti che non sono stati etichettati nei dati. La distanza dal centro del cluster dipende dalla forza dell'associazione. I punti dati possono appartenere ad un solo cluster. Un k più grande significa un gruppo più piccolo con più granularità allo stesso modo. Ad ogni cluster viene assegnata un'etichetta al punto dati.

In questa fase, è molto importante scalare il nostro dataset, in modo da standardizzare i dati, così da avere solo dati che variano nel range di -1 e 1.

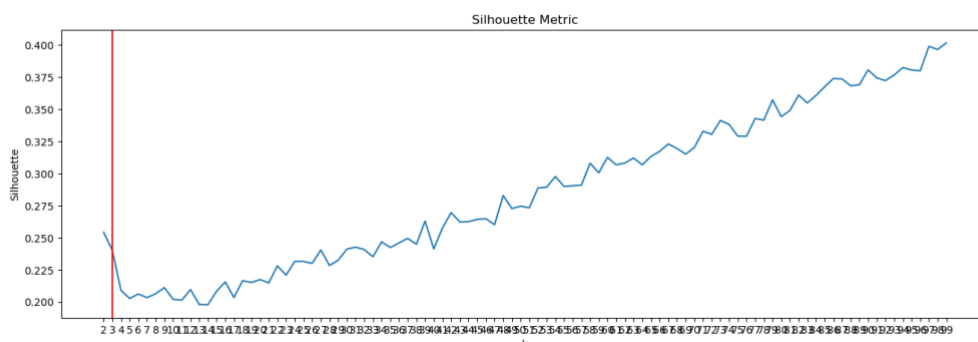


Una volta standardizzati i dati è importante scegliere un numero di clusters che si vuole identificare (es k=3), successivamente si selezionano casualmente 3 valori distinti che rappresenteranno i clusters iniziali. Come terzo punto si misura la distanza tra il primo punto nel dataset e i tre cluster iniziali, in base a chi ha distanza minore assegno quel dataset ad un cluster e faccio questa operazione per ogni punto.

In uno spazio a due dimensioni, per ogni cluster si trova il centroide (si fa la media della distanza dei punti assegnati al singolo cluster j).

Una situazione che potrebbe gravare molto sulle performance del nostro classificatore è la giusta scelta del valore K. Per ricercare il giusto K, utilizziamo una metrica chiamata **silhouette index**, in grado di quantificare quando un cluster è "puro":

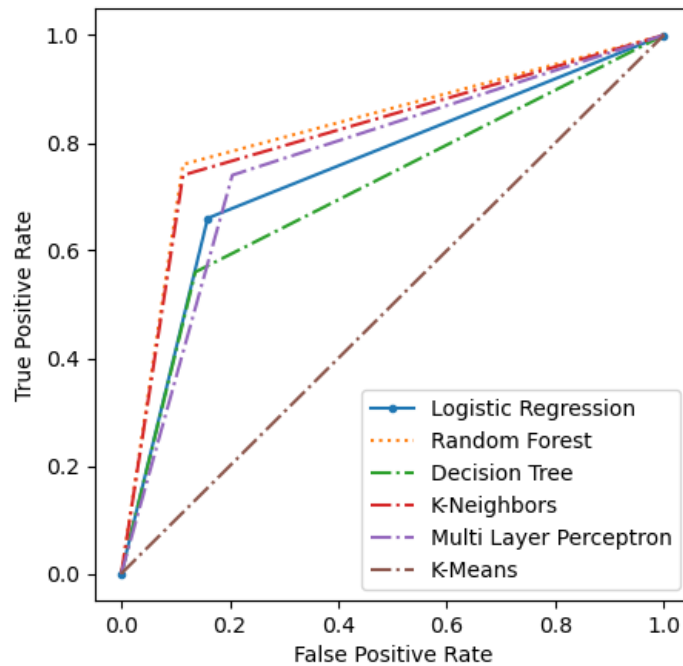
$$s = \frac{b - a}{\max(a, b)} [-1, 1]$$



Classification report con ottimizzazione del K:

	PRECISION	RECALL	F1 SCORE	SUPPORT
NONDEMENTED	0.50	0.30	0.37	44
DEMENTED	0.44	0.46	0.45	50
ACCURACY			0.38	94
MACRO AVG	0.31	0.25	0.27	94
WEIGHTED AVG	0.47	0.38	0.41	94

Conclusioni



Alla luce di quanto emerge dallo studio di questo grafico, possiamo notare come il *Random Forest* e *K-Neighbors* hanno un apprendimento rapido in una fase iniziale ed una crescita lineare dopo alcuni test, restituendo fin da subito un minore numero di falsi positivi.

La stessa curva di apprendimento è seguita anche dalla rete neurale e dall'algoritmo *Logistic Regression*, anche se questi ultimi tendono a sbagliare in una fase iniziale, e dopo poco apprendono correttamente restituendo sempre meno casi di falsi positivi.

L'algoritmo *Decision Tree* ha un apprendimento più lento rispetto agli altri citati finora.

Invece, l'algoritmo di apprendimento non supervisionato *K-Means* ha un apprendimento molto lento, e non molto preciso, dato che si inizia ad affinare alla fine dei nostri test.

Interfaccia a riga di comando (CLI)

Al momento, tramite console è possibile passare allenare la IA a creare un model, ricercare i paramenti migliori ed eseguire un test attraverso dei paramenti personali e ricevere dopo un poco un risultato che ci indicherà se il paziente è affetto da demenza senile oppure no.

```
Alzheimer prediction

1 -- Learning Supervised
2 -- Learning Unsupervised
3 -- Optimize parameters
4 -- Run test
5 -- Exit
Enter your choice:
```

Menù

Attraverso un semplice menù è possibile effettuare delle scelte:

1. Learning Supervised → Ci permetterà di allenare i diversi tipi di classificatori con apprendimento supervisionato
2. Learning Unsupervised → Allena i classificatori non supervisionati, precisamente K-Means
3. Optimize parameters → Tutti gli algoritmi vengono lanciati per ricercare i parametri migliori attraverso la GridSearch e K-Search

4. Run Test → Attraverso l'input utente, si potranno effettuare dei test. Al momento per semplicità di utilizzo, i dati vengono letti da un array statico.
5. Exit → Chiude il programma

Per poter effettuare una scelta sarà necessario indicare il numero dell'operazione che si vuole eseguire, ad esempio 1 per Learning Supervised.

Learning

Le funzioni Learning Supervised e Unsupervised si occuperanno di allenare i classificatori descritti precedentemente nella fase di analisi, restituendoci informazioni importanti che ci aiuteranno a comprendere quale classificatore converge meglio.

```
-- RANDOM FOREST CLASSIFIER
Train score: 1.0
Test score: 0.8191489361702128
Cross Validated Score: 0.8461538461538461
Standard Deviation: 0.0908838379258696
Variance: 0.008259871996135733
0-1 Loss: 0.18085106382978722
Accuracy: 0.8231818181818182

      precision    recall  f1-score   support

0         0.76      0.89      0.82         44
1         0.88      0.76      0.82         50

 accuracy          0.82
 macro avg         0.82
weighted avg         0.83
```

Optimize Params

Attraverso la **Grid Search** verranno trovati i parametri migliori per gli algoritmi di apprendimento supervisionato e il **K** per l'algoritmo non supervisionato.

```
BEST PARAMS:
- DTC: {'criterion': 'entropy', 'max_depth': 14} - 1.9423 seconds
- KNN: {'algorithm': 'kd_tree', 'n_neighbors': 5, 'p': 2, 'weights': 'distance'} - 0.3865 seconds
- RFC: {'bootstrap': True, 'criterion': 'gini', 'max_depth': 8, 'max_features': 'sqrt', 'n_estimators': 100} - 10.6772 seconds
- MLP: {'activation': 'relu'} - 0.4813 seconds
- LR: {'C': 5, 'penalty': 'l2'} - 0.1228 seconds
- kMeans: 0.1886 - 4.3946 seconds
-- Fine optimization end in 18.0047 seconds
```

Run Test

Il test viene eseguito caricando i classificatori addestrati precedentemente durante la fase di training.

Successivamente per ogni classificatore gli viene richiesto di effettuare un predict su un input di test che per semplicità viene caricato automaticamente.

Ogni predizione calcolata viene mostrata così da far scegliere all'utente a quale classificatore affidarsi, inoltre viene effettuata una media delle predizioni di ogni classificatore, visualizzata con la stringa "demented" o "non demented" in modo da facilitarne la lettura.

Parametri test

- Real: valori copiati dal file, il numero rappresenta la riga del file
- Inventato: valori supposti prendendo spunto dai diversi valori presenti nel file

```
# 'M/F', 'Age', 'EDUC', 'SES', 'MMSE', 'eTIV', 'nWBV', 'ASF'
xx = [[0, 87, 14, 2, 27, 1987, 0.696, 0.883]] # real: nondemented 1
xy = [[1, 80, 12, 2.0, 22.0, 1698, 0.701, 1.034]] # real: demented 4
xz = [[1, 76, 12, 2.0, 28.0, 1738, 0.713, 1.010]] # real: demented 3
zz = [[1, 99, 8, 2.0, 24.0, 1679, 0.800, 0.987]] # inventato
zx = [[1, 51, 15, 2.0, 24.0, 1679, 0.800, 0.987]] # inventato
# 'M/F', 'Age', 'EDUC', 'SES', 'MMSE', 'eTIV', 'nWBV', 'ASF'
```

Caso "demented"

```
-- Loaded trainend models in 0.0068 seconds

TEST: [1, 80, 12, 2.0, 22.0, 1698, 0.701, 1.034]

    DCT  KN  RaF  MLP  LR  KMS
0     1   1   1   1   1   0
The most predicted disease is demented

-- Predicted in 0.0344 seconds
```


Caso “non demented”

```
-- Loaded trainend models in 0.0102 seconds

TEST: [0, 87, 14, 2, 27, 1987, 0.696, 0.883]

    DCT  KN  RaF  MLP  LR  KMS
0     0   0    0    1   0    1
The most predicted disease is nondemented

-- Predicted in 0.0152 seconds
```

Caso inventato

```
-- Loaded trainend models in 0.0116 seconds

TEST: [1, 51, 15, 2.0, 24.0, 1679, 0.8, 0.987]

    DCT  KN  RaF  MLP  LR  KMS
0     1   0    1    0   1    0
The most predicted disease is nondemented

-- Predicted in 0.0306 seconds
```

Sviluppi Futuri

L’obiettivo iniziale del mio progetto era quello di creare un sistema automatico di identificazione dell’Alzheimer, in modo da dar un seguito al progetto in corso di sviluppo con la Professoressa De Carolis, ovvero una skill Alexa per aiutare le persone affette dalla perdita di memoria ad allenarsi costantemente attraverso micro-giochi interattivi.