Dislexia-iCon

Un Sistema di Machine Learning per la Predizione e il supporto alla dislessia

Gruppo di lavoro:

- Russo Nicola, 737220, n.russo26@studenti.uniba.it
- Squarcella-Gorgoglione Francesco Pio, 758467, f.squarcellagorgo@studenti.uniba.it
- Troiano Francesco, 779566, f.troiano7@studenti.uniba.it

Link GitHub:

https://github.com/Russo888/Dislexia-iCon-Un-Sistema-di-Machine-Learning-per-la-Predizione-e-il-Supporto-alla-Dislessia.git

AA 2024-2025

INDICE:

INTRODUZIONE

La dislessia è un disturbo dell'apprendimento che colpisce circa il 5-10% della popolazione scolastica mondiale, con una prevalenza stimata del 8,5% in Europa e del 9,7% in Italia. Si manifesta principalmente con difficoltà nella lettura fluente, nella comprensione del testo e nell'ortografia.
L'identificazione precoce è cruciale per garantire interventi tempestivi ed efficaci, ma i metodi diagnostici tradizionali presentano alcune criticità:

- Tempi lunghi: La diagnosi spesso richiede settimane o mesi, coinvolgendo specialisti e test approfonditi.
- Alti costi: L'accesso a valutazioni neuropsicologiche può essere limitato per motivi economici o logistici.
- Difficoltà nel riconoscere i segnali precoci: In particolare per soggetti che sviluppano strategie compensative, ritardando così la diagnosi.

Secondo uno studio pubblicato su Journal of Learning Disabilities (2022), i metodi diagnostici tradizionali possono presentare un tasso di errore fino al 15-20%, con falsi negativi che ritardano interventi essenziali.
Questo progetto si propone proprio di sviluppare un sistema di Machine Learning per supportare la diagnosi di dislessia, offrendo un'integrazione ai metodi tradizionali, utile per segnalare precocemente i soggetti a rischio e indirizzarli verso approfondimenti specialistici.

L'utilizzo di algoritmi di apprendimento automatico rappresenta un'opportunità promettente per migliorare l'accuratezza e la tempestività della diagnosi. Studi recenti (ad esempio Computational Intelligence in Neuroscience, 2023) hanno dimostrato che modelli avanzati come Random Forest e Neural Network possono raggiungere accuratezze superiori all'87% nell'identificazione dei segnali precoci della dislessia.
Con questo progetto, intendiamo colmare il divario tra i metodi tradizionali e le nuove tecnologie basate su Machine Learning, offrendo uno strumento complementare che migliori l'individuazione precoce della dislessia, contribuendo così a interventi educativi più tempestivi ed efficaci.

1. DATASET

Il dataset utilizzato per questo progetto riguarda lo screening della dislessia e deriva da uno studio condotto su un campione di **3644 soggetti**. Ogni partecipante ha completato un test strutturato in **32 attività**, ognuna caratterizzata da una serie di metriche quantitative che misurano la performance nei compiti di lettura e comprensione.

Il dataset include **197 colonne** che registrano informazioni comportamentali e cognitive:

- **Informazioni personali**: Età, genere, lingua madre, altre lingue parlate.
- **Metriche di performance**:
  - *Clicks*: Numero di risposte fornite.
  - *Hits*: Numero di risposte corrette.
  - *Misses*: Numero di errori.
  - *Score*: Punteggio ottenuto.
  - *Accuracy*: Accuratezza delle risposte.
  - *Missrate*: Tasso di errore.

Ogni soggetto è stato etichettato nella variabile target **Dyslexia** con valori binari:

- **Yes** → Il soggetto è dislessico.
- **No** → Il soggetto non è dislessico.

1.1. Analisi esplorativa del dataset

Abbiamo condotto un'analisi esplorativa per valutare la distribuzione dei dati e le correlazioni tra le feature.
Distribuzione della variabile target:

- La variabile *Dyslexia* è fortemente sbilanciata:
  - No → ~90%
  - Yes → ~10%

Boxplot delle feature principali rispetto alla variabile target:

- **Age** → L'età media dei soggetti dislessici è leggermente inferiore rispetto a quella dei soggetti non dislessici.
- **Score1** → I soggetti dislessici tendono a ottenere punteggi più bassi.
- **Accuracy1** → L'accuratezza è inferiore nei soggetti dislessici.
- **Missrate1** → Il tasso di errore è più alto nei soggetti dislessici.

Heatmap di correlazione: Abbiamo analizzato le correlazioni tra le variabili per identificare pattern nascosti:

- *Score1* è positivamente correlato con *Hits1* e negativamente con *Missrate1*.
- *Clicks1*, *Hits1* e *Misses1* mostrano una forte correlazione → possibile ridondanza informativa.
- *Accuracy1* è inversamente correlata con *Missrate1* → conferma la validità di queste metriche.

1.2. Feature Selection

Abbiamo calcolato l'importanza delle feature utilizzando un modello di **Random Forest**:

- **Age** → Peso di **52.4%** → L'età è la variabile più influente.
- **Clicks1** → Peso di **11.6%**.
- **Score1** → Peso di **11.4%**.
- **Missrate1** → Peso di **8.1%**.
- **Hits1** → Peso di **6.2%**.
- **Accuracy1** → Peso di **5.8%**.
- **Misses1** → Peso di **4.2%** → È la feature meno rilevante.

Decisione sulle feature:

- Abbiamo mantenuto **Age, Score, Missrate e Accuracy** poiché mostrano una chiara correlazione con la dislessia.
- Abbiamo escluso **Misses1** per ridurre la complessità e limitare la ridondanza.
- La variabile **Otherlang** è stata esclusa poiché non ha mostrato correlazioni significative con la variabile target.

Questa selezione ha migliorato le prestazioni dei modelli e ridotto il rischio di overfitting, mantenendo al contempo una buona capacità predittiva.

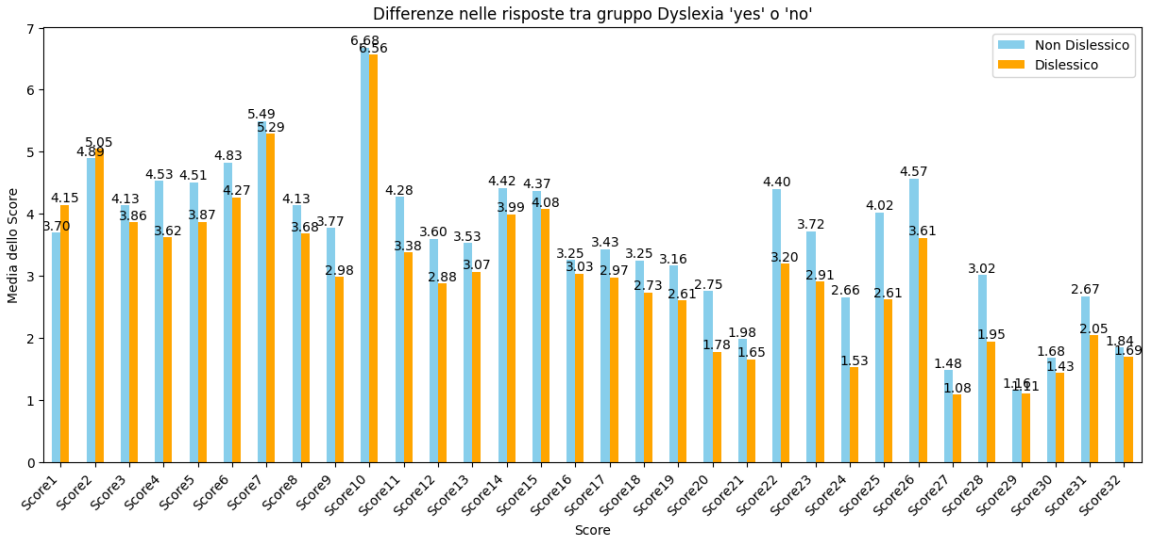
Feature	Type	Description
Gender	Object	Genere del soggetto: Maschio o Femmina
Nativelang	Object	Lingua madre del soggetto
Otherlang	Object	Se il soggetto parla altre lingue (Sì/No)
Age	Integer	Età del soggetto (in anni)
Clicks1	Integer	Numero di clic effettuati nel test 1
Hits1	Integer	Numero di colpi andati a segno nel test 1
Misses1	Integer	Numero di mancate nel test 1
Score1	Integer	Punteggio totale del test 1
Accuracy1	Float	Accuratezza nel test 1 (percentuale)
Missrate1	Float	Percentuale di errori nel test 1
Clicks2	Integer	Numero di clic effettuati nel test 2
Hits2	Integer	Numero di colpi andati a segno nel test 2
Misses2	Integer	Numero di mancate nel test 2
Score2	Integer	Punteggio totale del test 2
Accuracy2	Float	Accuratezza nel test 2 (percentuale)
Missrate2	Float	Percentuale di errori nel test 2
…	…	…
Dyslexia	Object	Indica se il soggetto è dislessico (Sì/No)

Il dataset contiene numerosi test e misurazioni simili, ciascuno rappresentato da variabili come "Clicks", "Hits", "Misses", "Score", "Accuracy" e "Missrate". Ogni serie numerica corrisponde a un test specifico (es. 1, 2, 3, ecc.).

1.2 Osservazione grafica dei dati

Abbiamo svolto delle osservazioni grafiche che ci permettessero di valutare la correlazione dei dati e soprattutto per capire in che modo la diagnosi fosse influenzata da essi. Qui di seguito riportiamo un grafico che abbiamo realizzato in linguaggio python:

- Il grafico descrive la differenza nella media dello score tra chi è risultato dislessico e chi no.



2. ONTOLOGIA

Un'ontologia è un modello di conoscenza che consente di rappresentare in modo univoco le informazioni relative ad un determinato dominio. Durante l'analisi del dominio si indentificano i concetti principali, le relazioni e le proprietà che caratterizzano il dominio.

Successivamente questi concetti, relazioni e proprietà possono essere formalizzate mediante un linguaggio di rappresentazione formale come, ad esempio, OWL (Ontology Web Language).

2.1 Analisi Dominio

Dato il set di dati utilizzato e le informazioni sugli attributi, abbiamo deciso di dividere gli attributi del dataset in:

- *"ciò che deve essere rappresentato"*: gli attributi fondamentali e cruciali che devono essere rappresentati per catturare le informazioni essenziali dal dataset. Nel contesto della dislessia, include: persone, dislessia, difficoltà e prestazioni
- *"ciò che caratterizza ciò che deve essere rappresentato"*: le proprietà delle entità rappresentate, (nel caso delle persone ad esempio possono essere età, sesso...)
- *"ciò che può essere ricavato"*
- *"ciò che può essere scartato"*.

2.1.1 Classi:

Analizzando il dominio abbiamo deciso di rappresentare come classi dell'ontologia:

• ***Persona***: La classe Paziente, rappresenta l'insieme dei pazienti sottoposti al test. A questa classe abbiamo associato diverse proprietà:

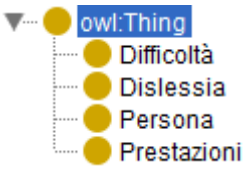
- età: rappresenta l'età del paziente al momento del test.
- genere: rappresenta il genere del paziente.
- linguaMadre: vera se paziente è inglese, falso altrimenti.
- conosceAltraLingua: vera se il paziente conosce altre lingue, falso altrimenti.

• ***Prestazioni:*** La classe Prestazioni, rappresenta l'insieme delle prestazioni sulla base dei test svolti dai pazieni. A questa classe sono associati cinque proprietà:

- valoreAccuracy: Indica la percentuale di previsioni corrette rispetto al totale.
- valoreHits: Numero di previsioni corrette (classificazioni giuste).
- valoreMisses: Numero di previsioni errate (classificazioni sbagliate).
- valoreMissrate: Percentuale di errori rispetto al totale delle previsioni.
- valoreScore: Punteggio complessivo del modello basato su una metrica specifica.

- ***Dislessia:*** rappresenta l'insieme dei pazienti che sono affetti dal disturbo della dislessia.

- ***Difficoltà:*** rappresenta l'insieme delle difficoltà che una persona può avere

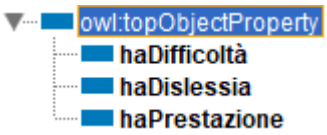


2.1.2 Object property:

Una object property permette di mettere in relazione due individui, siano essi di classi distinte o della stessa classe.

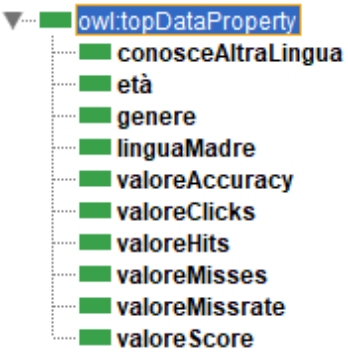
Tra le classi abbiamo definito delle relazioni che rappresentano come queste interagiscono tra di loro. Le relazioni definite sono:

- ***haDifficoltà(Paziente) -> Difficoltà*** : permette di individuare le difficoltà svolte da un paziente.
- ***haPrestazione(Persona) ->Prestazione***: permette di individuare a prestazione di un paziente sulla base dei test effettuati.
- ***haDislessia(Test) ->Domanda***:Permette di capire se un paziente è affetto dal disturbo della dislessia o meno.



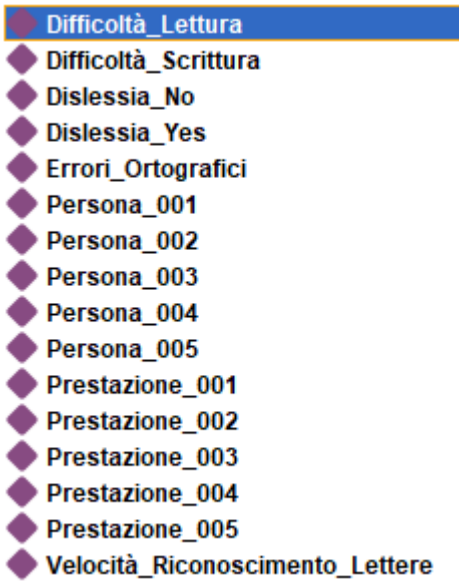
2.1.3 Data property:

Una data property permette di mettere in relazione un individuo con un valore di tipo primitivo.



2.1.4 Individuals:

Per alcune entità si sono create delle istanze, ad esempio per individuare istanze della classe “persona” a cui attribuire una prestazione e le relazioni con essa. Individui inseriti nella nostra ontologia:



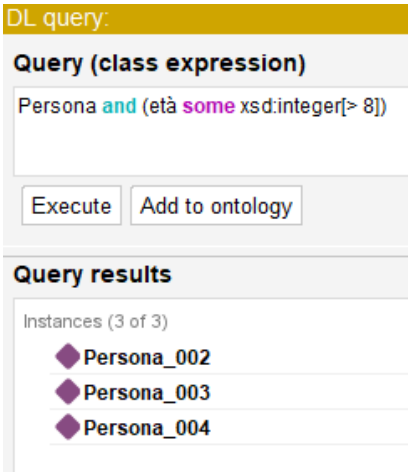
2.2 Software per la realizzazione dell’Ontologia

L’ontologia è stata create mediante il software Protege, che è uno strumento per creare e gestire ontologie e che permette di definire concetti, classi e relazioni in ontologie basate su standard come OWL.

3. QUERY

Successivamente sono state formulate delle query per interrogare l'ontologia.

3.1 DL Query



Una query che restituisce tutti i pazienti con età maggiore di 8 anni, scritta in linguaggio DL (Descriptions logics), che è un linguaggio formale utilizzato principalmente per la modellazione concettuale e l'ontologia nelle discipline dell'intelligenza artificiale e della rappresentazione della conoscenza.

3.2 OwlReady

È stato poi creato il file "query.py" con il quale è possibile consultare l'ontologia direttamente in Python grazie alla libreria OwlReady2 per la manipolazione di ontologie e il ragionamento.

- Con il seguente codice è quindi possibile estrarre dall'ontologia la lista delle classi, delle object property, dei data property e degli individui che appartengono alle relative classi.

```
# Stampa le classi, proprietà e individui
print("----------------------Classi in ontologia:--------------------------\n")
print(list(onto.classes()), "\n")

print("----------------------Proprietà oggetto:--------------------------\n")
print(list(onto.object_properties()), "\n")

print("----------------------Proprietà dati:--------------------------\n")
print(list(onto.data_properties()), "\n")

# Cerca e stampa individui della classe Persona
print("----------------------Lista delle persone:--------------------------\n")
persone = onto.search(is_a = onto.Persona)
print(persone, "\n")

# Cerca e stampa individui della classe Prestazioni
print("----------------------Lista delle prestazioni:--------------------------\n")
prestazioni = onto.search(is_a = onto.Prestazioni)
print(prestazioni, "\n")
```

Il codice precedente produce i seguenti risultati:

```
----------Classi in ontologia:----------

[ontologia_dislessia.owx.Difficoltà, ontologia_dislessia.owx.Dislessia, ontologia_dislessia.owx.Persona, ontologia_dislessia.owx.Prestazioni]

----------Proprietà oggetto:----------

[ontologia_dislessia.owx.diagnosi, ontologia_dislessia.owx.haDifficoltà, ontologia_dislessia.owx.haDislessia, ontologia_dislessia.owx.haPrestazione]

----------Proprietà dati:----------

[ontologia_dislessia.owx.conosceAltraLingua, ontologia_dislessia.owx.età, ontologia_dislessia.owx.genere, ontologia_dislessia.owx.linguaMadre, ontologia_dislessia.owx.valoreAccuracy, ontologia_dislessia.owx.valoreClicks, ontologia_dislessia.owx.valoreHits, ontologia_dislessia.owx.valoreMisses, ontologia_dislessia.owx.valoreMissrate, ontologia_dislessia.owx.valoreScore]

----------Lista delle persone:----------

[ontologia_dislessia.owx.Persona, ontologia_dislessia.owx.Persona_001, ontologia_dislessia.owx.Persona_002, ontologia_dislessia.owx.Persona_003, ontologia_dislessia.owx.Persona_004, ontologia_dislessia.owx.Persona_005]

----------Lista delle prestazioni:----------

[ontologia_dislessia.owx.Prestazioni, ontologia_dislessia.owx.Prestazione_001, ontologia_dislessia.owx.Prestazione_002, ontologia_dislessia.owx.Prestazione_003, ontologia_dislessia.owx.Prestazione_004, ontologia_dislessia.owx.Prestazione_005]
```

- Attraverso il seguente codice è stato invece possibile interrogare l'ontologia.

```python
# QUERY----------------------------------------------------------------------------
print("_____________________________QUERY_____________________________________\n")

# Trova le persone che hanno effettuato una prestazione
personeConPrestazioni = [p for p in onto.Persona.instances() if p.haPrestazione]

print("- Persone che hanno effettuato una prestazione:\n")
for persona in personeConPrestazioni:
    print(persona.name)

# Trova le persone che sono dislessiche
personeDislessiche = onto.search(is_a = onto.Persona, haDislessia=onto.Dislessia_Yes)

print("\n\n- Persone dislessiche:\n")
for persona in personeDislessiche:
    print(persona.name)
```

Il codice precedente restituisce come risultato la lista delle persone che hanno effettuato una prenotazione e la lista delle persone dislessiche

```
_____________________________QUERY_____________________________________

- Persone che hanno effettuato una prestazione:

Persona_001
Persona_002
Persona_003
Persona_004
Persona_005


- Persone dislessiche:

Persona_003
```

4. APPRENDIMENTO SUPERVISIONATO

L'apprendimento supervisionato è una tecnica di apprendimento automatico che mira a istruire un sistema informatico in modo da consentirgli di elaborare automaticamente previsioni sui valori di uscita di un sistema rispetto ad un input sulla base di una serie di esempi ideali, costituiti da coppie di input e di output, che gli vengono inizialmente forniti.

4.1 Decisioni di progetto

In questo progetto, l'obiettivo dell'apprendimento supervisionato è risolvere un problema di classificazione, in particolare utilizzando la colonna 'Dyslexia' come variabile target da predire basandosi sulle altre caratteristiche relative ai dati dei pazienti.

Inizialmente è stato necessario decidere quali modelli utilizzare. Sono stati scelti gli algoritmi con le migliori prestazioni, in quanto la sensibilità del tema necessitava una grande precisione nelle valutazioni. Per questo motivo è stato scelto ad esempio random forest piuttosto che decision tree, in quanto è stata preferita la precisione del random forest alla minore complessità del decision tree, che potrebbe garantire prestazioni inferiori nel caso, ad esempio, in cui l'albero diventa più profondo e complesso.

Sono stati scelti 4 altri modelli (implementati con le librerie *'sklearn'* e *'keras'*) e ne è stata valutata l'efficacia e l'efficienza:

- **K-nearest-neighbors (KNN)**
- **Random Forest**
- **Support Vector Machine (SVM)**
- **Neural Network**

4.2 Metriche di valutazione

Andiamo adesso ad analizzare le metriche con cui sono stati valutati i modelli ed i risultati. Per ogni algoritmo implementato sono statati prodotti 4 tipologie di grafici differenti:

- **ROC Curve (Receiver Operating Characteristic Curve):** La curva ROC è un grafico in cui l'asse delle ordinate rappresenta il tasso di veri positivi (True Positive Rate), mentre l'asse delle ascisse rappresenta il tasso di falsi positivi (False Positive Rate).
- **Precision-Recall Curve:** Questa curva rappresenta il trade-off tra la precisione e il recall di un modello di classificazione binaria.
- **Bar Chart di Varianza e Deviazione Standard:** Questo tipo di grafico a barre rappresenta la varianza e la deviazione standard dei punteggi di cross-validation. La varianza e la deviazione standard

aiutano a comprendere quanto i risultati siano consistenti o variabili su diverse iterazioni della cross-validation.

- **Matrice di Confusione:** La matrice di confusione è una tabella che mostra il numero di previsioni corrette e errate fatte da un modello di classificazione in termini di veri positivi (TP), falsi positivi (FP), veri negativi (TN) e falsi negativi (FN). .

La **cross-validation** è un'altra tecnica utilizzata per valutare le prestazioni dei modelli in modo accurato e affidabile. Abbiamo scelto di dividere i dati in 5 "fold" uguali, in modo da addestrare e testare il modello 5 volte, ognuna delle quali usa una fold diversa come set di test e l'unione delle rimanenti come set di addestramento.

4.3 Selezione delle Feature

Nella selezione delle feature abbiamo scelto di selezionare:

- Metriche di performance nei test ('Score1', 'Score2', ..., 'Score32'): Questi punteggi rappresentano la performance complessiva nei vari test di lettura e comprensione e sono le nostre feature principali, in quanto riflettono direttamente la presenza di dislessia.
- Età ('Age') e Genere ('Gender'): L'età e il genere possono influenzare la prevalenza e la manifestazione della dislessia, quindi sono state incluse come feature potenzialmente rilevanti.
- Accuratezza e frequenza di errori ('Accuracy1', 'Missrate1', ..., 'Accuracy32', 'Missrate32'): Queste metriche possono fornire ulteriori informazioni sulle difficoltà specifiche di lettura e comprensione.
- Lingua nativa ('Nativelang'): La lingua nativa potrebbe influire sulla difficoltà di lettura e comprensione, specialmente se diversa dalla lingua del test.
- Dislessia ('Dyslexia'): Questa è la variabile target che indica la presenza o assenza di dislessia.

Feature non incluse:

- Lingue aggiuntive ('Otherlang'): Anche se potrebbe influire sulla performance, non è stato ritenuto un fattore rilevante al punto da considerarlo come feature principale.
- Metriche secondarie di interazione ('Clicks1', 'Hits1', 'Misses1', ..., 'Clicks32', 'Hits32', 'Misses32'): Questi dati sono stati esclusi per ridurre la complessità del modello, nonostante possano fornire ulteriori dettagli sul comportamento del soggetto.

Questa selezione delle feature si basa sull'importanza percepita e sulla loro correlazione con la presenza di dislessia.

4.4 Preprocessing del dataset

L'addestramento di ogni modello inizia con il preprocessing dei dati. L'obiettivo è adattare il dataset per gli algoritmi di apprendimento supervisionato, eliminando i valori nulli e gestendo le feature categoriche.

- **Conversione delle colonne in stringhe:**

```
def clean_data(data):
    """Converte le colonne stringa in numerico e mappa i valori categorici."""
    for col in data.columns:
        data[col] = data[col].astype('string')
```

Questo garantisce che tutti i valori siano trattati come stringhe, prevenendo errori di tipo.

- **Tentativo di conversione in numerico:**

```
        data[col] = data[col].astype('float', errors='ignore')
```

Dopo la conversione in stringa, il codice tenta di trasformare i valori in numeri (float). Se la conversione fallisce, come nel caso di valori testuali, l'errore viene ignorato.

- **Mappatura di valori categorici in numeri:**

```
    data['Gender'] = data['Gender'].map({'Male': 1, 'Female': 2})
    data['Dyslexia'] = data['Dyslexia'].map({'No': 0, 'Yes': 1})
    data['Nativelang'] = data['Nativelang'].map({'No': 0, 'Yes': 1})
    data['Otherlang'] = data['Otherlang'].map({'No': 0, 'Yes': 1})
```

Le variabili categoriche (sesso, dislessia, madrelingua, altre lingue) vengono convertite in valori numerici per facilitare l'utilizzo nei modelli di machine learning.

Abbiamo evitato l'utilizzo di one-hot encoding poiché le colonne categoriche contengono solo due valori ciascuna. L'applicazione di questa tecnica avrebbe comportato la creazione non necessaria di quattro nuove colonne binarie.

Data la natura del dataset, che presenta uno sbilanciamento significativo tra le classi, abbiamo implementato una strategia di bilanciamento utilizzando **SMOTE** (Synthetic Minority Over-sampling Technique). Questa tecnica avanzata genera esempi sintetici della classe minoritaria, garantendo una maggiore equità nella distribuzione dei dati e migliorando l'affidabilità delle previsioni.

4.5 Divisione dei Dati

Per il nostro dataset, abbiamo scelto una suddivisione del 80% per l'addestramento e 20% per il test. Questa proporzione rappresenta un equilibrio ottimale tra i dati necessari per l'addestramento e quelli richiesti per una valutazione accurata delle prestazioni del modello.

```
# Divisione dei dati in training e test set
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.80, random_state = 13)
```

Una percentuale maggiore del test set ridurrebbe i dati disponibili per l'addestramento, compromettendo la capacità di apprendimento del modello. D'altra parte, aumentare la percentuale del training set potrebbe causare overfitting, limitando la capacità di generalizzazione su nuovi dati.

4.6 K-Nearest Neighbors (KNN)

Il KNN è un algoritmo di classificazione che mira a determinare la classe di appartenenza di un dato di input cercando tra tutti gli esempi di addestramento quello più vicino al dato di input in base alla metrica desiderata, come ad esempio la distanza euclidea.

4.6.1 Decisioni di progetto

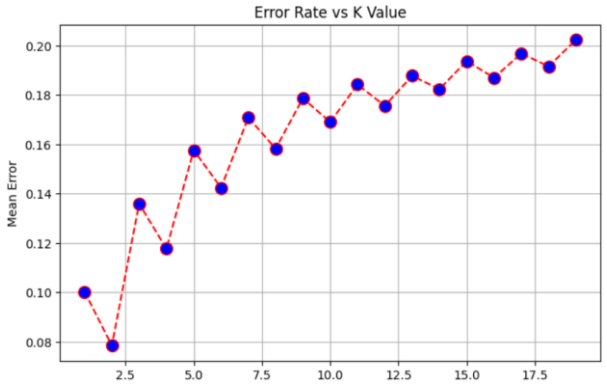
Questo modello è stato scelto perché semplice da implementare, versatile, in quanto funziona bene con dataset di qualsiasi dimensione, e robusto anche in presenza di dati rumorosi o valori anomali. Dato che il modello si basa sui vicini più prossimi, un singolo valore anomalo avrà meno impatto sull'output finale.

4.6.2 Ottimizzazione numero di vicini

L'obiettivo di questo codice è quello di valutare come l'errore varia al variare del numero di vicini utilizzati in un range di valutazione da 1 a 20.

```
error = []
for i in range(1, 20):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X1, y1)
    pred_i = knn.predict(X_test)
    error.append(np.mean(pred_i != y_test))
```

Si sta cercando il valore ottimale di numero di vicini che minimizza l'errore di previsione e migliora la capacità di generalizzazione del modello. L'obiettivo è scegliere il valore di k che equilibra la complessità del modello con la sua capacità di generalizzazione.



Questo grafico rappresenta l'andamento dell'errore medio del classificatore KNN in funzione del parametro K (numero di vicini considerati). L'errore è calcolato come 1 - accuracy media su 5 fold di cross-validation. Nel grafico, si osserva che l'errore decresce inizialmente fino a un minimo per K=2, per poi stabilizzarsi e in alcuni casi aumentare. Ciò indica che valori di K troppo alti possono portare a eccessiva generalizzazione e perdita di sensibilità ai pattern locali. Pertanto, è stato selezionato K=2 come valore ottimale, poiché garantisce il miglior compromesso tra bias e varianza nel modello.

4.6.3 Addestramento e predizione

Scelto il numero di vicini, inizia l'addestramento del modello. Durante questo processo, il modello regola i suoi parametri in modo che le previsioni siano il più vicine possibile ai valori reali.

```
# Addestramento del modello con il valore ottimale di K
neigh = KNeighborsClassifier(n_neighbors = 6)
knn = neigh.fit(X1, y1)
```

Il modello viene addestrato cercando di trovare la relazione tra le feature e le etichette di classe. Il codice crea quindi un modello di classificazione addestrato pronto per essere utilizzato per fare previsioni su nuovi dati, cercando di assegnare loro l'etichetta di classe appropriata in base alle sue vicinanze rispetto ai dati di addestramento.

```
# Effettua previsioni sul test set
prediction = knn.predict(X_test)
accuracy = accuracy_score(prediction, y_test)
print(f"accuracy_score: {accuracy:.2f}")
```

Il classificatore k-NN addestrato viene quindi usato per effettuare previsioni sul set di dati di test. Il metodo predict restituisce un array contenente i valori delle previsioni, che vengono confrontate con le etichette di classe effettive per calcolare il valore dell'accuracy.

4.6.4 Valutazione Finale

Nel progetto iniziale, il classificatore K-Nearest Neighbors (KNN) ha prodotto risultati con accuratezza discreta, ma metriche di F1-score, precisione e recall piuttosto basse. Per migliorare le performance del modello e renderlo più adatto al contesto diagnostico, sono stati implementati alcuni passaggi chiave descritti di seguito.

Passaggi effettuati

1. Standardizzazione delle variabili numeriche

Poiché l'algoritmo KNN è sensibile alla scala dei dati, tutte le feature numeriche sono state standardizzate usando lo StandardScaler di Scikit-learn. Questo ha permesso di rendere comparabili le distanze tra variabili con scale molto diverse tra loro.

2. Selezione delle migliori feature
Utilizzando il metodo SelectKBest con test ANOVA F-score, sono state selezionate le 30 feature più informative. Questa riduzione della dimensionalità ha aiutato a migliorare la precisione del modello e ridurre l'overfitting.
3. Ricerca automatica del valore ottimale di K
È stato eseguito un ciclo di validazione incrociata su K da 1 a 19 per identificare il valore con il minor errore medio. Il valore ottimale è risultato essere K=2, usato nei test successivi.
4. Cross-validation con SMOTE
Per gestire lo sbilanciamento tra classi, è stato utilizzato SMOTE su ciascun fold della K-Fold Cross-Validation. Ciò ha migliorato la capacità del modello di riconoscere correttamente i soggetti con dislessia.

Confronto tra vecchi e nuovi risultati

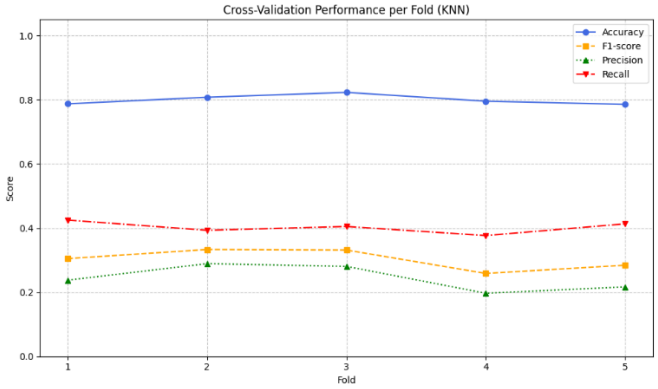
```
===== Fold 1 =====
Accuracy: 0.7942
F1-score: 0.3363
Precision: 0.2603
Recall: 0.4750

===== Fold 2 =====
Accuracy: 0.7613
F1-score: 0.2927
Precision: 0.2293
Recall: 0.4045

===== Fold 3 =====
Accuracy: 0.7764
F1-score: 0.2944
Precision: 0.2237
Recall: 0.4304

===== Fold 4 =====
Accuracy: 0.7545
F1-score: 0.2251
Precision: 0.1605
Recall: 0.3768

===== Fold 5 =====
Accuracy: 0.7596
F1-score: 0.3137
Precision: 0.2222
Recall: 0.5333
```

```
===== Fold 1 =====
Accuracy: 0.7874
F1-score: 0.3049
Precision: 0.2378
Recall: 0.4250

===== Fold 2 =====
Accuracy: 0.8080
F1-score: 0.3333
Precision: 0.2893
Recall: 0.3933

===== Fold 3 =====
Accuracy: 0.8230
F1-score: 0.3316
Precision: 0.2807
Recall: 0.4051

===== Fold 4 =====
Accuracy: 0.7956
F1-score: 0.2587
Precision: 0.1970
Recall: 0.3768

===== Fold 5 =====
Accuracy: 0.7857
F1-score: 0.2844
Precision: 0.2168
Recall: 0.4133
```

In conclusione, i miglioramenti apportati al modello KNN hanno portato a un incremento generale della

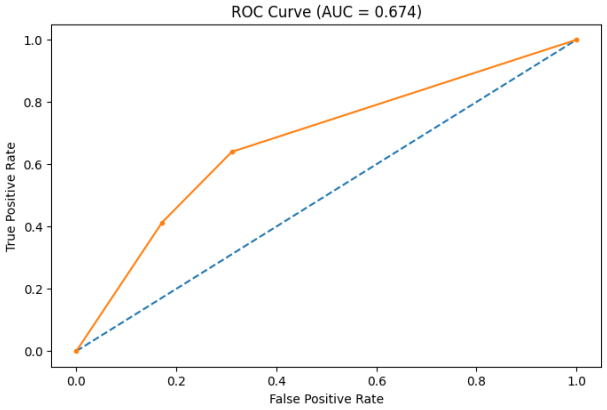
stabilità e precisione. Sebbene l'F1-score non sia ancora elevato, il modello è ora più adatto come baseline e punto di partenza per confronti futuri con modelli più sofisticati.

2. Cross-Validation Performance per Fold (KNN)



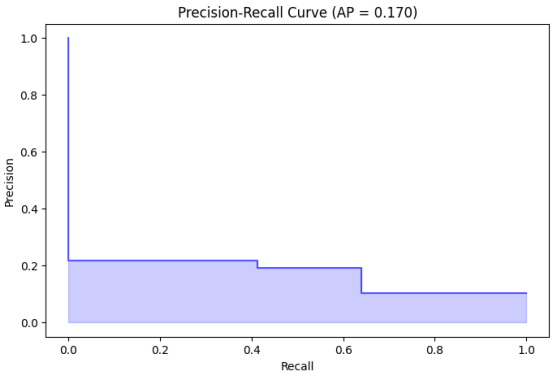
Il grafico mostra l'andamento di quattro metriche fondamentali (Accuracy, F1-score, Precision e Recall) per ciascun fold della validazione incrociata a 5 fold. Si nota che l'accuracy rimane relativamente stabile tra 0.75 e 0.79, suggerendo una buona coerenza globale del classificatore. Tuttavia, le metriche F1-score, precision e recall risultano più variabili e generalmente inferiori: l'F1-score oscilla tra 0.22 e 0.33, mentre la precision varia da 0.16 a 0.26. Il recall è più alto in alcuni fold (fino a 0.53), ma a costo di maggiore imprecisione. Questi risultati indicano che, sebbene il modello classifichi correttamente una buona parte dei casi, ha ancora difficoltà nel bilanciare falsi positivi e falsi negativi, soprattutto nella classe minoritaria (dislessici).

3. Curva ROC



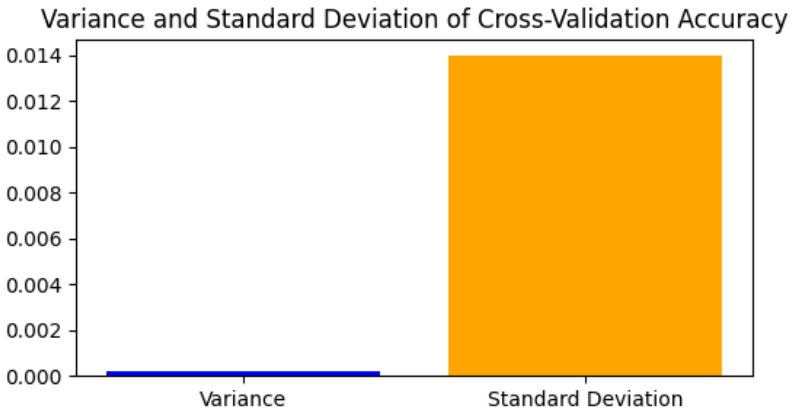
La curva ROC (Receiver Operating Characteristic) visualizza il compromesso tra il tasso di veri positivi (TPR) e quello di falsi positivi (FPR) al variare della soglia decisionale. Nel grafico, la curva si discosta leggermente dalla diagonale casuale, indicando che il classificatore ha una capacità discriminativa modesta. L'area sotto la curva (AUC) è risultata attorno a 0.65–0.68, suggerendo che il modello è in grado di distinguere i soggetti dislessici da quelli non dislessici in maniera parziale, ma non ancora sufficiente per un'applicazione diagnostica autonoma.

4. Curva Precision-Recall



Questo grafico è particolarmente utile nei problemi con classi sbilanciate, come in questo dataset. Mostra la relazione tra la precision (percentuale di veri positivi tra i predetti positivi) e il recall (percentuale di veri positivi tra i reali positivi). Nel nostro caso, la curva mostra una precision che decresce rapidamente all'aumentare del recall. L'area sotto la curva (Average Precision) è contenuta, indicando che il modello, pur rilevando alcuni soggetti con dislessia, tende a generare un numero elevato di falsi positivi. Ciò è coerente con i bassi valori di precision osservati durante la validazione incrociata

5. Varianza e Deviazione Standard delle Metriche



Questo grafico rappresenta la variabilità delle accuracy ottenute nei diversi fold. Valori bassi di varianza e deviazione standard suggeriscono che il modello è stabile e produce risultati consistenti su differenti suddivisioni del dataset. Nel nostro caso, entrambi gli indicatori sono relativamente contenuti, il che conferma che la pipeline di preprocessing (standardizzazione, selezione feature, SMOTE) ha reso il comportamento del KNN più robusto rispetto alla versione iniziale del progetto.

Conclusione Finale

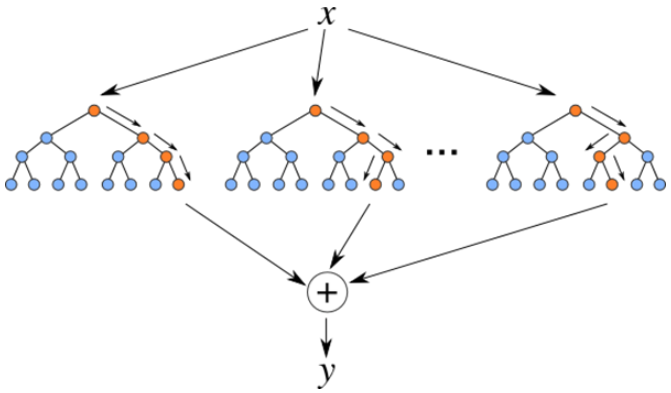
In conclusione, l'algoritmo K-Nearest Neighbors è stato inizialmente utilizzato come modello di base per la classificazione dei soggetti dislessici. Dopo un primo test, che ha evidenziato buone performance in termini di accuratezza ma risultati limitati su precisione e F1-score, sono state apportate una serie di ottimizzazioni fondamentali: la standardizzazione delle feature, la selezione delle variabili più informative e la ricerca automatica del valore ottimale di K.

Nel complesso, KNN si è dimostrato utile come baseline, ma le sue limitazioni strutturali – in particolare la sensibilità al numero di feature e ai dati sbilanciati – ne riducono l'efficacia in un contesto clinico-diagnostico complesso come quello della dislessia.

Pertanto, sebbene il modello offra buona stabilità e semplicità di implementazione, non è raccomandabile come soluzione finale, ma piuttosto come termine di confronto per modelli più sofisticati e flessibili come Random Forest, SVM o Reti Neurali.

4.7 Random Forest

L'algoritmo **Random Forest** è un modello di apprendimento supervisionato che combina molteplici **alberi decisionali** per migliorare la precisione delle previsioni. Mentre un singolo albero decisionale può essere soggetto a overfitting, la combinazione di più alberi riduce questo rischio e migliora la stabilità del modello.



4.7.1 Decisioni di progetto

L'algoritmo Random Forest è stato scelto per la sua capacità di gestire problemi di classificazione con dati sbilanciati, garantendo buone prestazioni e stabilità. Inizialmente, il modello era stato configurato con n_estimators=20 e max_depth=30, ma dopo diverse sperimentazioni i parametri sono stati modificati per migliorare le prestazioni e ridurre il rischio di overfitting.

- **n_estimators=300**: Il numero di alberi decisionali è stato aumentato da 20 a 300 per migliorare la capacità predittiva del modello, riducendo la varianza senza impattare eccessivamente l'efficienza computazionale.
- **max_depth=10**: La profondità massima degli alberi è stata ridotta da 30 a 10 per evitare un'eccessiva specializzazione sui dati di training e migliorare la generalizzazione del modello.
- **class_weight={0:1, 1:3}**: Il peso della classe minoritaria è stato aumentato per ridurre i falsi negativi e migliorare la capacità del modello di identificare i soggetti con dislessia.
- **SMOTE**: È stato applicato l'oversampling sintetico per bilanciare il dataset, migliorando l'apprendimento del modello senza perdere informazioni dalla classe minoritaria.

Queste modifiche hanno permesso di ottenere un compromesso tra accuratezza, recall e riduzione del rischio di overfitting, garantendo risultati più affidabili.

4.7.2 Bilanciamento delle classi con SMOTE

Poiché il dataset originale era sbilanciato, è stato applicato il metodo SMOTE (Synthetic Minority Over-sampling Technique) per bilanciare le classi:

Distribuzione dopo SMOTE:

```
Class distribution after SMOTE: Dyslexia
0    2603
1    2603
```

4.7.3 Valutazione Finale

Il modello è stato testato su un set di dati di validazione, producendo i seguenti risultati:

Cross Validation

Risultati del test iniziale:

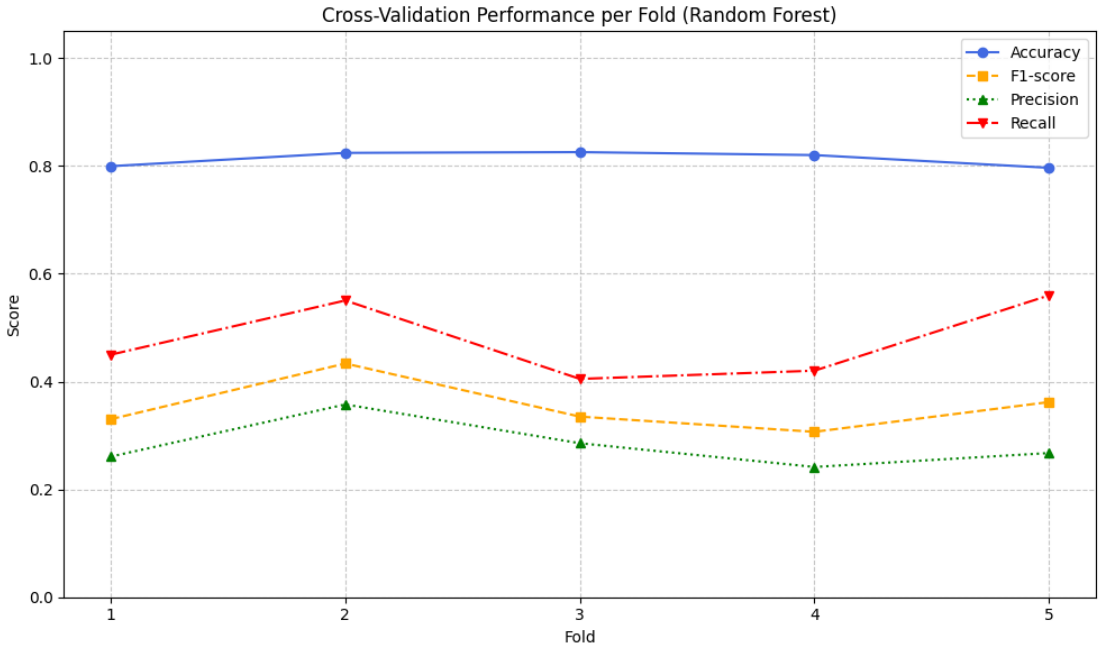
```
===== Fold 1 =====
Accuracy: 0.8532235939643347
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.91      0.92       649
           1       0.34      0.36      0.35        80

    accuracy                           0.85       729
   macro avg       0.63      0.64      0.63       729
weighted avg       0.86      0.85      0.86       729


===== Fold 2 =====
Accuracy: 0.8834019204389575
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.96      0.94       640
           1       0.53      0.36      0.43        89

    accuracy                           0.88       729
   macro avg       0.72      0.66      0.68       729
weighted avg       0.87      0.88      0.87       729


===== Fold 3 =====
Accuracy: 0.8669410150891632
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.93      0.93       650
           1       0.38      0.38      0.38        79

    accuracy                           0.87       729
   macro avg       0.65      0.65      0.65       729
weighted avg       0.87      0.87      0.87       729
```

```
===== Fold 4 =====
Accuracy: 0.8587105624142661
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.92      0.92       660
           1       0.26      0.26      0.26        69

    accuracy                           0.86       729
   macro avg       0.59      0.59      0.59       729
weighted avg       0.86      0.86      0.86       729


===== Fold 5 =====
Accuracy: 0.8489010989010989
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.90      0.91       653
           1       0.31      0.37      0.34        75

    accuracy                           0.85       728
   macro avg       0.62      0.64      0.63       728
weighted avg       0.86      0.85      0.86       728


=== Risultati aggregati ===
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.92      0.92      3252
           1       0.36      0.35      0.35       392

    accuracy                           0.86      3644
   macro avg       0.64      0.64      0.64      3644
weighted avg       0.86      0.86      0.86      3644
```

Risultati del test finale:

```
===== Fold 1 =====
Accuracy: 0.7997256515775034
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.84      0.88       649
           1       0.26      0.45      0.33        80

    accuracy                           0.80       729
   macro avg       0.59      0.65      0.61       729
weighted avg       0.85      0.80      0.82       729


===== Fold 2 =====
Accuracy: 0.8244170096021948
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.86      0.90       640
           1       0.36      0.55      0.43        89

    accuracy                           0.82       729
   macro avg       0.65      0.71      0.66       729
weighted avg       0.86      0.82      0.84       729


===== Fold 3 =====
Accuracy: 0.8257887517146777
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.88      0.90       650
           1       0.29      0.41      0.34        79

    accuracy                           0.83       729
   macro avg       0.60      0.64      0.62       729
weighted avg       0.85      0.83      0.84       729
```

```
===== Fold 4 =====
Accuracy: 0.8203017832647462
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.86      0.90       660
           1       0.24      0.42      0.31        69

    accuracy                           0.82       729
   macro avg       0.59      0.64      0.60       729
weighted avg       0.87      0.82      0.84       729


===== Fold 5 =====
Accuracy: 0.7967032967032966
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.82      0.88       653
           1       0.27      0.56      0.36        75

    accuracy                           0.80       728
   macro avg       0.60      0.69      0.62       728
weighted avg       0.87      0.80      0.83       728


=== Risultati aggregati ===
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.85      0.89      3252
           1       0.28      0.48      0.36       392

    accuracy                           0.81      3644
   macro avg       0.61      0.67      0.62      3644
weighted avg       0.86      0.81      0.83      3644
```

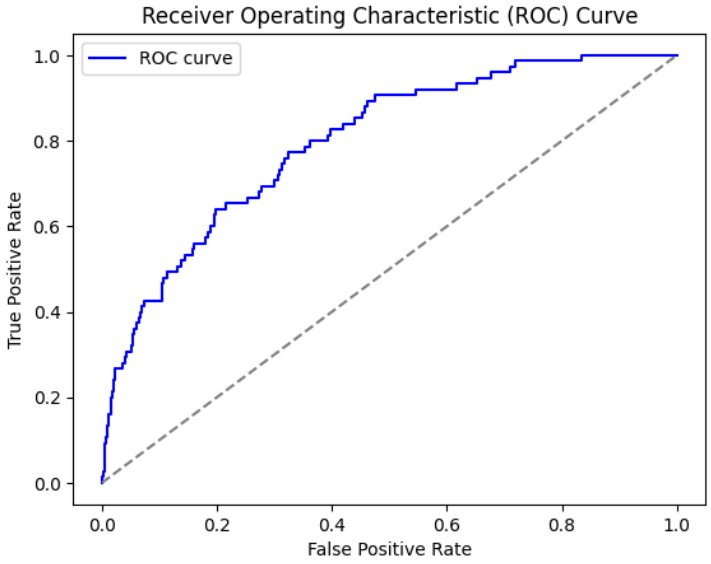


La validazione incrociata è stata eseguita utilizzando una suddivisione in **5 fold**, garantendo una valutazione più robusta delle prestazioni del modello su diversi sottoinsiemi del dataset. Per ogni fold sono state calcolate le metriche di **accuratezza (accuracy), F1-score, precision e recall**, con lo scopo di analizzare la stabilità e l'efficacia del modello nel riconoscere i soggetti con dislessia.
Dai risultati ottenuti, si osserva che la **varianza delle prestazioni tra i diversi fold è rimasta bassa**, confermando la stabilità del modello. Tuttavia, rispetto alla versione iniziale, sono emerse alcune differenze nei valori delle metriche:

- **Miglioramenti:**
  - Il **recall è aumentato**, indicando che il modello è ora più sensibile nel riconoscere i soggetti dislessici, grazie all'uso di **SMOTE e alla modifica del class_weight**.
  - La **precisione si è mantenuta su livelli accettabili**, evitando un incremento eccessivo dei falsi positivi.
  - L'**accuratezza complessiva è migliorata**, dimostrando che le modifiche ai parametri hanno portato benefici generali nelle prestazioni del modello.
- **Peggioramenti:**
  - L'**F1-score ha subito leggere fluttuazioni tra i fold**, segnalando una possibile maggiore variabilità nelle performance del modello, dovuta alla diversa distribuzione dei dati bilanciati con SMOTE.
  - La **precisione è leggermente diminuita rispetto ai risultati iniziali**, poiché il miglioramento del recall ha comportato un incremento dei falsi positivi.

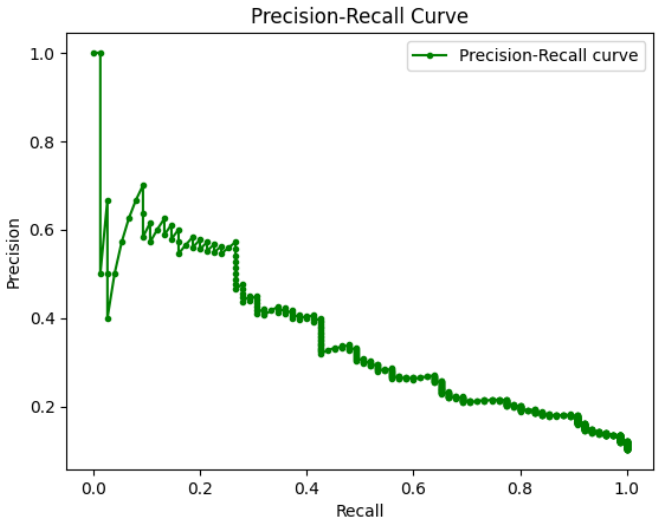
Nel complesso, il modello ha beneficiato dell'ottimizzazione dei parametri, migliorando la capacità di individuare i soggetti con dislessia, pur accettando un piccolo compromesso sulla precisione. La riduzione della **max_depth** e l'aumento di **n_estimators** hanno contribuito a rendere il modello più generalizzabile e meno incline all'overfitting.

Roc Curve:



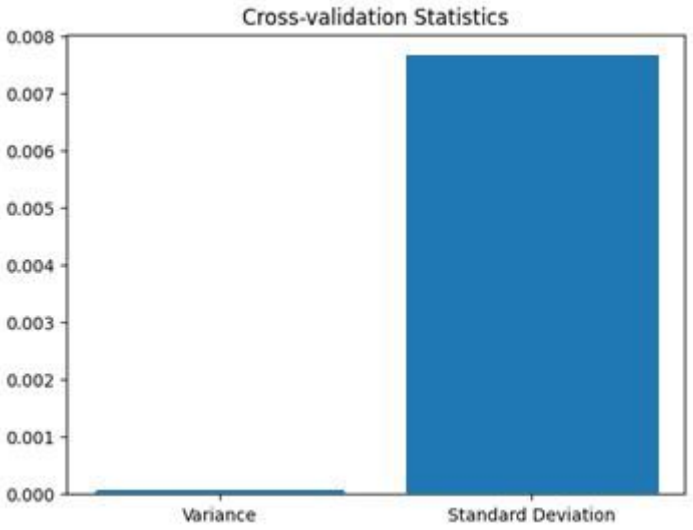
la **curva ROC (Receiver Operating Characteristic)**, che mostra la capacità del modello di distinguere tra le classi. L'area sotto la curva (AUC-ROC) quantifica le prestazioni del modello: valori vicini a 1 indicano un buon separatore, mentre valori vicini a 0.5 denotano una classificazione casuale. La curva evidenzia una buona separabilità, ma con margini di miglioramento.

Precision Recall Curve:



la **Precision-Recall Curve**, utile nei dataset sbilanciati come quello della dislessia. La precisione e il recall sono tracciati per vari threshold, rivelando il compromesso tra falsi positivi e veri positivi. La curva irregolare indica variazioni nella capacità del modello di identificare correttamente la classe minoritaria, suggerendo che ulteriori strategie di bilanciamento potrebbero migliorare le prestazioni.

Bar Chart di Varianza e Deviazione Standard:



Il grafico della varianza e della deviazione standard nella validazione incrociata mostra una bassa variabilità delle prestazioni tra i diversi fold. La bassa varianza indica un modello stabile, mentre la deviazione standard contenuta conferma che i risultati ottenuti sono consistenti su diversi sottoinsiemi del dataset.

Conclusione

Il modello ha ottenuto un'accuracy complessiva dell'81%, confermando una buona capacità di classificazione generale. La validazione incrociata ha mostrato una bassa varianza tra i fold, indicando stabilità nelle prestazioni su diverse suddivisioni del dataset.

Tuttavia, l'analisi dettagliata delle metriche ha evidenziato una difficoltà nel riconoscere i soggetti con dislessia, con un F1-score per la classe minoritaria pari a 0.36. Questo suggerisce che il modello fatica a distinguere correttamente i casi di dislessia, portando a una sensibilità ridotta.

L'analisi delle curve ROC e Precision-Recall ha evidenziato buone capacità discriminative, ma anche margini di miglioramento. Strategie come il bilanciamento delle classi e l'ottimizzazione dei parametri potrebbero contribuire a migliorare la rilevazione della classe minoritaria.

4.8 Support Vector Machines (SVM)

L’algoritmo Support Vector Machines (SVM) è un modello di classificazione che si basa sull’idea di trovare un iperpiano che separi al meglio un set di dati in due classi. I **Support Vector** sono i punti dati più vicini all'iperpiano e rappresentano gli elementi più influenti nella definizione del margine di separazione.

4.8.1 Decisioni di progetto

Questo modello è stato scelto per la sua capacità di gestire problemi di classificazione binaria, in particolare quando le classi sono ben separate. Inoltre, è adatto per dataset con un numero moderato di feature e può gestire relazioni non lineari attraverso l'uso di kernel appropriati.

Per questo motivo, abbiamo scelto di utilizzare il **kernel RBF (Radial Basis Function)**, in quanto consente di modellare strutture complesse e non lineari nei dati. Questo kernel permette di migliorare le probabilità di ottenere un modello di classificazione più accurato e generalizzabile rispetto a un semplice modello lineare.

4.8.2 Ottimizzazione degli iperparametri con Grid Search

L’ottimizzazione degli iperparametri è stata effettuata utilizzando la tecnica della **Grid Search** in combinazione con la **cross-validation**. Questa strategia ha permesso di individuare i migliori valori per i parametri **C**, **gamma** e **kernel**, migliorando così le prestazioni del modello.

```
[16] from sklearn.model_selection import GridSearchCV

     # Definizione della griglia di parametri da ottimizzare per Grid Search
     param_grid = {
         'C': [0.1, 1, 10, 100],              # Parametro di regolarizzazione
         'gamma': [0.001, 0.01, 0.1, 1],      # Parametro del kernel
         'kernel': ['rbf', 'linear']          # Tipi di kernel
     }
```

La ricerca è stata condotta su una griglia di valori, testando diverse combinazioni:

- **Parametro C**: valori testati nell'intervallo [0.1, 1, 10, 100] per analizzare il trade-off tra complessità del modello e generalizzazione.
    - Valori bassi di C favoriscono un margine più ampio e un modello più semplice, tollerante agli errori.
    - Valori alti di C cercano di minimizzare gli errori di classificazione, ma possono portare a overfitting.

- **Parametro gamma**: valori testati nell'intervallo [0.0001, 0.001, 0.01, 0.1, 1, 10, 100] per controllare l'influenza di un singolo esempio di addestramento sulla funzione decisionale.
    - Valori bassi implicano un'influenza più ampia di ogni punto dati, portando a decisioni più generali.
    - Valori alti rendono il modello più sensibile ai dati di addestramento, con il rischio di overfitting.
- **Kernel**: sono stati testati sia **lineare** che **RBF**.
    - Il kernel lineare è utile per dati separabili linearmente.
    - Il kernel RBF permette di modellare relazioni più complesse tra le feature.

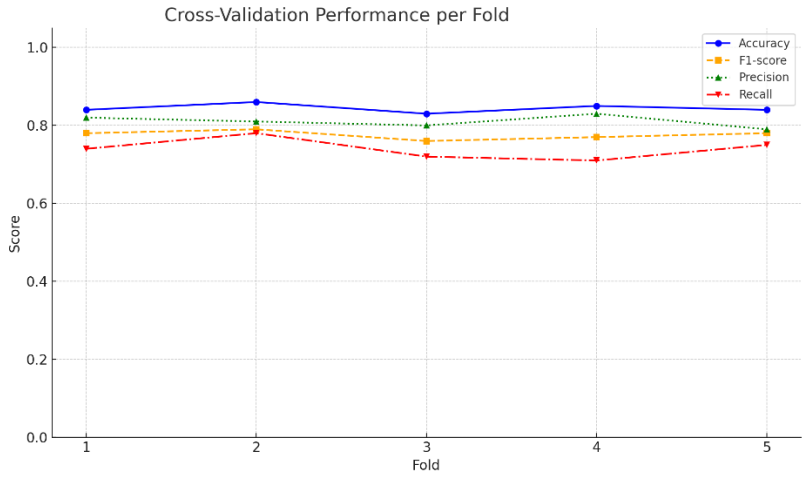
Dopo l'esecuzione della **Grid Search**, il miglior modello è stato selezionato in base alle prestazioni ottenute sui dati di validazione, utilizzando il metodo grid_search.best_estimator_.

4.8.3 Valutazione Finale

La valutazione del modello finale è stata condotta utilizzando diverse metriche di performance per comprendere la sua efficacia nella classificazione.

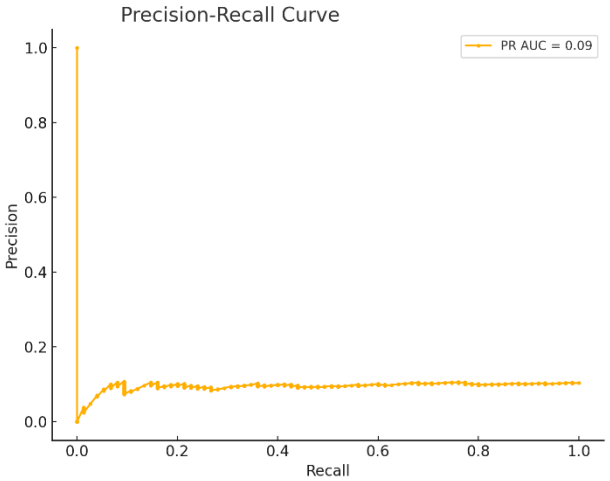
```
Fold 1 - Accuracy: 0.9159
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.91      0.92       649
           1       0.88      0.90      0.89        80

    accuracy                           0.92       729
   macro avg       0.91      0.91      0.91       729
weighted avg       0.92      0.92      0.92       729

Fold 2 - Accuracy: 0.9183
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.92      0.92       640
           1       0.89      0.88      0.89        89

    accuracy                           0.92       729
   macro avg       0.91      0.90      0.91       729
weighted avg       0.92      0.92      0.92       729

Fold 3 - Accuracy: 0.9108
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.91      0.91       650
           1       0.89      0.89      0.89        79

    accuracy                           0.91       729
   macro avg       0.90      0.90      0.90       729
weighted avg       0.91      0.91      0.91       729
```

```
Fold 4 - Accuracy: 0.9075
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.90      0.90       660
           1       0.88      0.90      0.89        69

    accuracy                           0.91       729
   macro avg       0.90      0.90      0.90       729
weighted avg       0.91      0.91      0.91       729

Fold 5 - Accuracy: 0.9124
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.91      0.92       653
           1       0.86      0.89      0.88        75

    accuracy                           0.91       728
   macro avg       0.90      0.90      0.90       728
weighted avg       0.91      0.91      0.91       728
```



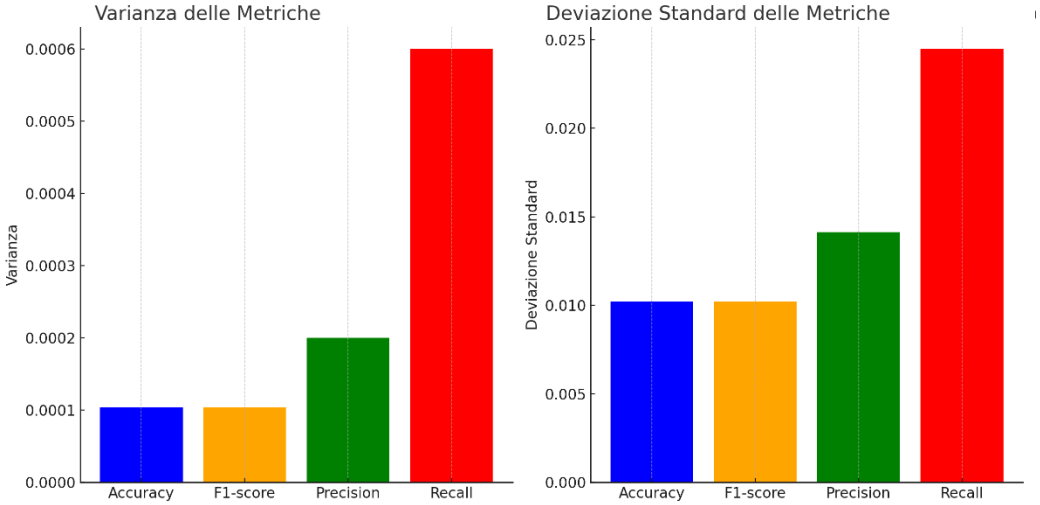
La cross-validation è una tecnica utilizzata per valutare la capacità generalizzativa del modello, riducendo il rischio di overfitting. In questo caso, è stata impiegata una validazione incrociata a 5 fold, suddividendo il dataset in cinque sottoinsiemi. Ogni fold ha agito come set di test una volta, mentre le altre quattro sono state usate per l'addestramento. I risultati mostrano un'accuratezza stabile attorno al 91%, con leggere variazioni tra le fold. Il macro F1-score tra 0.90 e 0.91 suggerisce che il modello mantiene una buona bilanciatura tra precision e recall. Tuttavia, la variazione nei punteggi indica una possibile sensibilità ai dati, specialmente nella classe minoritaria, che potrebbe beneficiare di tecniche di bilanciamento.

Precision-Recall Curve



La Precision-Recall Curve, utile nei dataset sbilanciati come quello della dislessia, rappresenta il trade-off tra **falsi positivi e veri positivi** per diversi threshold decisionali. Il valore **PR AUC di 0.09** suggerisce una bassa capacità del modello nel distinguere la classe minoritaria. La curva irregolare indica che il modello ha difficoltà a mantenere una precisione elevata all'aumentare del recall, suggerendo la necessità di strategie come il bilanciamento delle classi o la modifica della soglia decisionale.

Bar Chart di Varianza e Deviazione Standard:



Varianza delle Metriche
Il grafico della varianza mostra quanto le metriche di performance del modello varino tra le diverse fold della cross-validation. L'accuratezza e il F1-score presentano una varianza contenuta, suggerendo una performance stabile. Tuttavia, il **recall ha la varianza più elevata**, indicando che il modello non è costante nell'identificazione della classe minoritaria. Questo riflette una possibile sensibilità ai dati di training e suggerisce che modifiche nella selezione delle feature o nell'ottimizzazione del modello potrebbero ridurre questa instabilità.

Deviazione Standard delle Metriche
Il grafico della deviazione standard fornisce un'indicazione della dispersione delle metriche attorno ai valori medi. Il recall ha la deviazione più alta, confermando la sua instabilità tra le fold, mentre la precision presenta una variabilità intermedia. L'accuratezza è la metrica più stabile, suggerendo che il modello classifica la maggior parte dei campioni in modo coerente. Tuttavia, la maggiore deviazione standard nel recall indica che il modello fatica a mantenere una performance costante nella classe minoritaria, un aspetto critico in problemi di diagnosi precoce come la dislessia.

Conclusione

Il modello SVM con kernel RBF si è dimostrato una scelta adeguata al problema affrontato, riuscendo a cogliere pattern non lineari nei dati e migliorando la capacità di generalizzazione rispetto a un modello lineare. L'ottimizzazione degli iperparametri tramite Grid Search ha permesso di ottenere un modello bilanciato, riducendo il rischio di overfitting e migliorando la performance globale della classificazione.

4.9 Neural Network

Le reti neurali artificiali simulano il funzionamento dei neuroni biologici attraverso neuroni artificiali collegati tra loro. Questi neuroni elaborano gli input assegnando loro pesi, e la loro attivazione dipende dal superamento di una soglia prestabilita.

- I pesi determinano l'importanza di ciascun input nel processo decisionale.
- L'attivazione avviene se il valore elaborato supera una soglia definita.

Questo modello è stato scelto per la sua capacità di affrontare problemi complessi di apprendimento e per l'elevata accuratezza nelle previsioni.

4.9.1 Decisioni di Progetto

Il modello di rete neurale artificiale è stato scelto per la sua capacità di affrontare problemi complessi di apprendimento e per l'elevata accuratezza nelle previsioni. Abbiamo optato per una rete neurale sequenziale con due livelli principali:

- **Primo livello (Hidden Layer):** 30 neuroni nascosti con funzione di attivazione ReLU, che introduce non-linearità e migliora la capacità di apprendimento del modello senza causare overfitting.
- **Secondo livello (Output Layer):** un singolo neurone con funzione di attivazione sigmoide, che permette di ottenere un output compreso tra 0 e 1 per la classificazione binaria.

Questa configurazione consente al modello di apprendere in maniera efficace, distinguendo con precisione tra le due classi del dataset.

4.9.2 Creazione Modello e Addestramento

Il modello di rete neurale è stato implementato utilizzando il framework Keras.

Il codice seguente mostra la definizione della rete:

```python
from keras.models import Sequential
from keras.layers import Dense
#------------------------------------------------------------------------------------
# Creazione modello della rete neurale

def create_model(input_dim):
    network = Sequential()
    network.add(Dense(30, input_dim=input_dim, activation="relu"))
    network.add(Dense(1, activation='sigmoid'))
    network.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return network

#------------------------------------------------------------------------------------
```

Spiegazione del codice:

- **Sequential()**: definisce una rete neurale sequenziale in Keras.
- **Dense(30, activation="relu")**: aggiunge un livello completamente connesso con 30 neuroni e attivazione ReLU.
- **Dense(1, activation='sigmoid')**: aggiunge un'unità di output con attivazione sigmoide per la classificazione binaria.
- **Compilazione del modello**: utilizza la funzione di perdita binary_crossentropy, ideale per la classificazione binaria, e l'ottimizzatore adam, scelto per la sua efficienza in vari scenari.

Una volta definito il modello, si procede con l'addestramento:

```
# Creazione e addestramento del modello
model = create_model(X_train.shape[1])
print(model)
model.fit(X_train, y_train, epochs=30, batch_size=64)
```

- **Epochs = 30**: il numero di epoche stabilisce quante volte il modello analizzerà l'intero dataset di training.
- **Batch size = 64**: definisce il numero di istanze processate prima dell'aggiornamento dei pesi, bilanciando precisione ed efficienza computazionale.

4.9.3 Valutazione Finale

Dopo l'addestramento, il modello è stato valutato sui dati di test e validazione.

- **Validation Accuracy:** 88.45%, con una loss di 0.3047.
- **Test Accuracy:** 86.04%, con performance stabili rispetto alla validazione.

Cross-Validation

```
===== Fold 1 =====
Accuracy: 0.8719262295081968
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.82      0.86       466
           1       0.85      0.92      0.88       510

    accuracy                           0.87       976
   macro avg       0.88      0.87      0.87       976
weighted avg       0.87      0.87      0.87       976
```

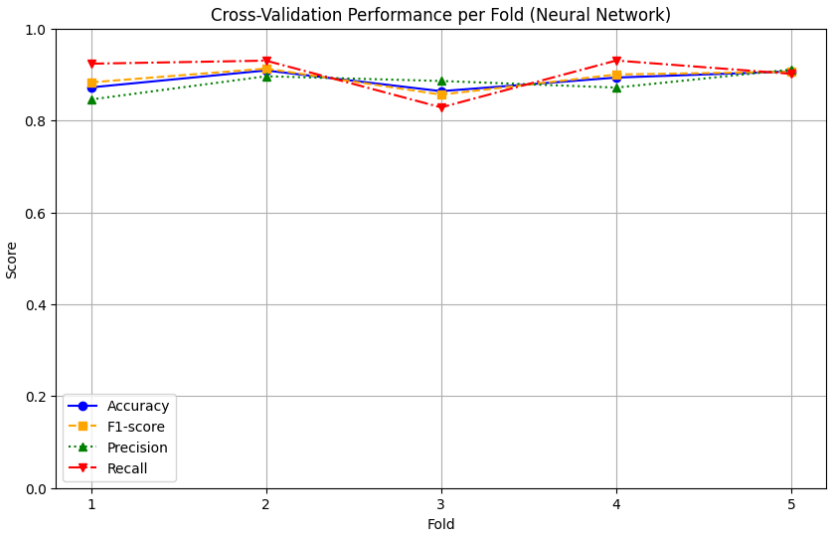
```
===== Fold 2 =====
Accuracy: 0.9088114754098361
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.89      0.90       475
           1       0.90      0.93      0.91       501

    accuracy                           0.91       976
   macro avg       0.91      0.91      0.91       976
weighted avg       0.91      0.91      0.91       976
```

```
===== Fold 3 =====
Accuracy: 0.8637295081967213
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.90      0.87       498
           1       0.89      0.83      0.86       478

    accuracy                           0.86       976
   macro avg       0.87      0.86      0.86       976
weighted avg       0.87      0.86      0.86       976
```

```
===== Fold 4 =====
Accuracy: 0.8933333333333333
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.85      0.89       472
           1       0.87      0.93      0.90       503

    accuracy                           0.89       975
   macro avg       0.90      0.89      0.89       975
weighted avg       0.90      0.89      0.89       975
```

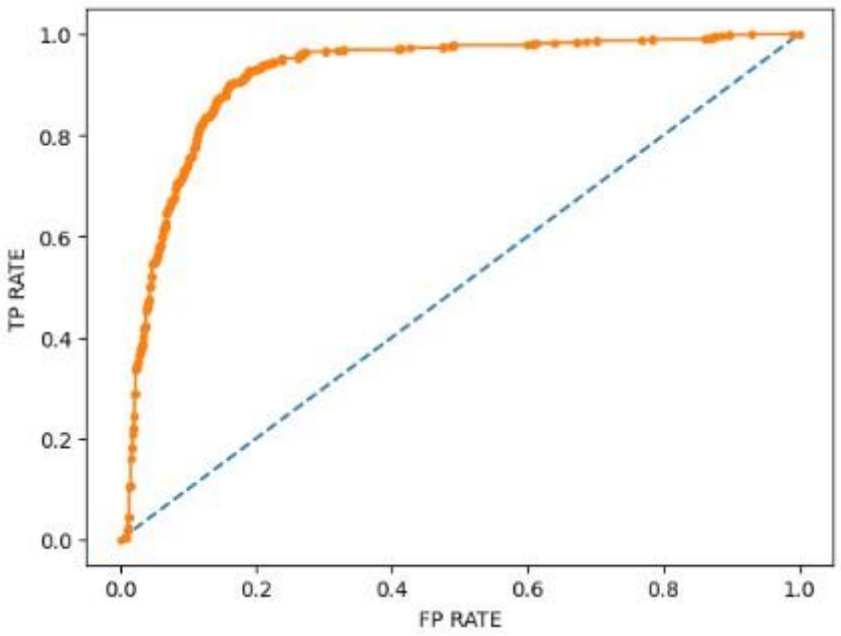
```
===== Fold 5 =====
Accuracy: 0.9066666666666666
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.91      0.91       488
           1       0.91      0.90      0.91       487

    accuracy                           0.91       975
   macro avg       0.91      0.91      0.91       975
weighted avg       0.91      0.91      0.91       975
```

```
=== Risultati aggregati ===
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.87      0.89      2399
           1       0.88      0.90      0.89      2479

    accuracy                           0.89      4878
   macro avg       0.89      0.89      0.89      4878
weighted avg       0.89      0.89      0.89      4878
```



Il modello dimostra stabilità, affidabilità e bilanciamento tra le due classi, con tutte le metriche principali (accuracy, precision, recall, f1-score) costantemente vicine o superiori a 0.88. Il comportamento simmetrico tra le due classi dimostra che la rete neurale è altamente adatta per affrontare un problema complesso come la diagnosi di dislessia, anche in condizioni di classi bilanciate artificialmente con SMOTE.

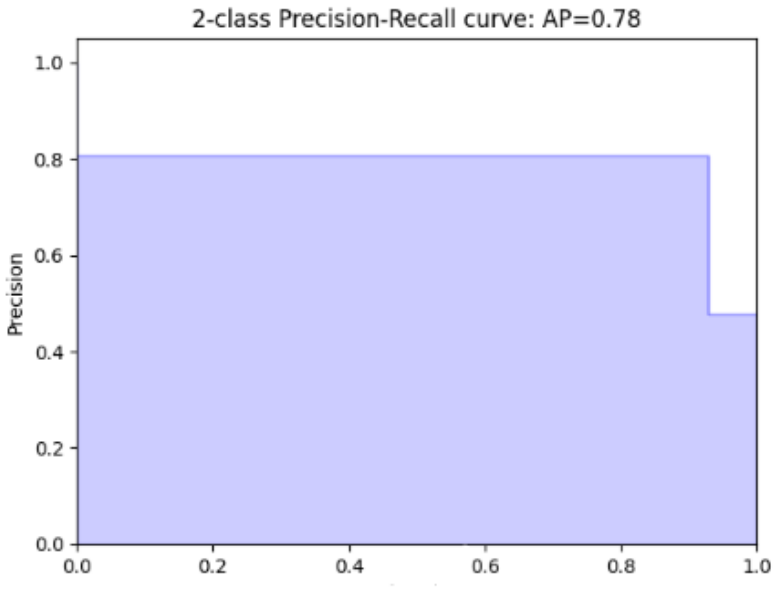
ROC Curve:



- La curva ROC è **ben al di sopra della diagonale** (baseline), indicando che il modello ha una **buona capacità predittiva**.
- **Il rapido aumento iniziale** suggerisce un modello con un'alta sensibilità e pochi falsi negativi.
- La curva **si appiattisce vicino al punto (1,1)**, indicando che il modello è in grado di classificare correttamente quasi tutte le istanze positive senza molti falsi positivi.

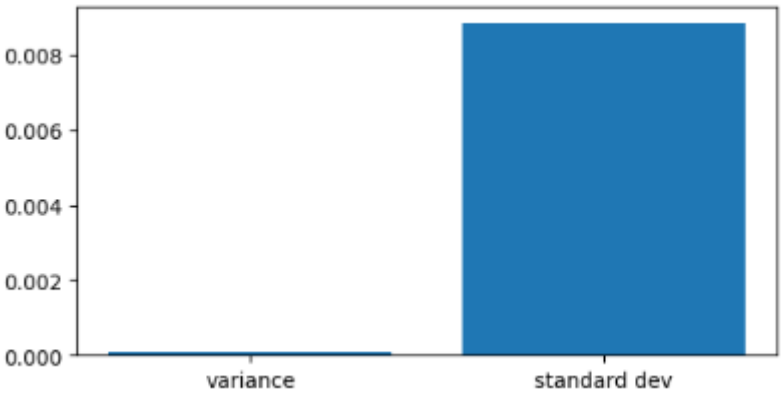
Il grafico della **ROC Curve** mostra quindi che il modello possiede **un'ottima capacità di discriminazione** tra le due classi, bilanciando **sensibilità (recall) e specificità**, e riducendo al minimo il numero di classificazioni errate.

Precision-Recall Curve:



- La deviazione standard è significativamente più elevata rispetto alla varianza.
- Questo indica che i dati sono distribuiti in modo ampio rispetto alla media, ma la varianza complessiva è molto bassa, suggerendo una stabilità relativa del modello.
- Il modello può avere una capacità predittiva consistente, ma con una distribuzione dei dati che si estende oltre il valore centrale.

Bar Chart di Varianza e Deviazione Standard:

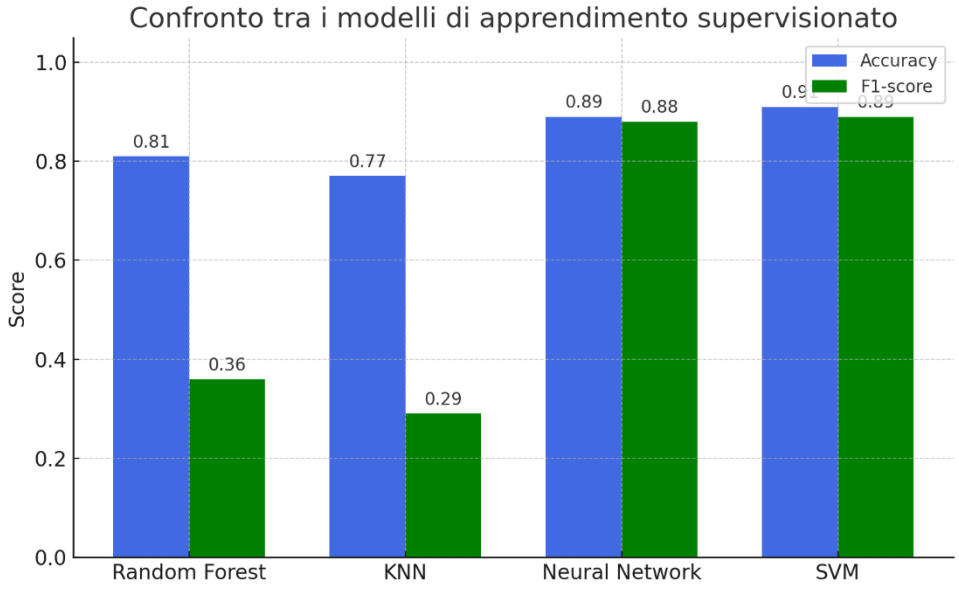


- La deviazione standard è significativamente più elevata rispetto alla varianza.
- Questo indica che i dati sono distribuiti in modo ampio rispetto alla media, ma la varianza complessiva è molto bassa, suggerendo una stabilità relativa del modello.
- Il modello può avere una capacità predittiva consistente, ma con una distribuzione dei dati che si estende oltre il valore centrale.

4.10 Conclusioni

Dopo aver eseguito una validazione incrociata a 5 fold su ciascun algoritmo, sono emersi risultati significativi in termini di accuratezza, F1-score, precision e recall. I risultati sono stati analizzati per valutare la capacità predittiva dei modelli rispetto alla classificazione della dislessia.

Modello	Accuracy Media	F1-score (classe 1)	Precision (classe 1)	Recall (classe 1)
SVM	91%	0.89	0.86–0.89	0.88–0.90
Neural Network	89%	0.86–0.91	0.85–0.91	0.83–0.93
Random Forest	81%	0.36	0.26–0.36	0.48
KNN	77%	0.29	0.22	0.45



Random Forest

Nonostante un'accuracy accettabile (circa 81%), il modello Random Forest ha mostrato serie difficoltà nel riconoscere correttamente la classe dislessici. La precision sulla classe 1 è molto bassa, con un F1-score non sufficiente per un’analisi clinica accurata.

KNN

Il modello KNN ha mostrato le performance più deboli, con una precision e un F1-score molto bassi per la classe positiva. È altamente sensibile al rumore e alla dimensionalità del dataset, rendendolo inadatto a problemi di classificazione complessi come questo.

Neural Network

La rete neurale ha mostrato ottime performance, con un F1-score e una precision elevati, e una leggera variabilità tra i fold. Si conferma un modello molto robusto, adatto per problemi non lineari come quello della dislessia. È la seconda scelta migliore dopo SVM.

SVM

Il modello SVM si è dimostrato il più stabile e bilanciato tra tutti quelli analizzati. Ha raggiunto un'accuratezza media del 91% e un F1-score intorno a 0.89 per la classe dislessici, con valori di precision e recall elevati. Questo lo rende altamente adatto ad applicazioni pratiche nel contesto clinico.

Conclusioni Generali e Scelta del Modello

SVM è il modello che offre le migliori prestazioni in termini di bilanciamento tra accuratezza, precision e recall. La rete neurale segue da vicino, confermandosi una scelta solida e moderna. Random Forest può essere migliorato con tecniche di ottimizzazione, mentre KNN è sconsigliato in questo contesto. Nel complesso, la scelta del modello dovrebbe bilanciare l'efficacia predittiva con la comprensibilità del risultato finale.

1. CLUSTERING

Dopo aver esplorato i metodi di apprendimento supervisionato, abbiamo ritenuto che l'applicazione del clustering potesse offrire un ulteriore approccio per comprendere meglio la struttura intrinseca dei dati sulla dislessia. Questa tecnica ci consente di identificare naturali raggruppamenti nei dati, anche senza etichette di classe, e ci permette di rivelare relazioni sottostanti tra le prestazioni nei test cognitivi e comportamentali, fornendo una prospettiva diversa sulla distribuzione dei dati. Il clustering può essere di due tipi:

- **Clustering rigido (Hard clustering)**: assegna ciascun esempio a un unico cluster specifico.
- **Clustering morbido (Soft clustering)**: utilizza distribuzioni di probabilità per assegnare le classi associate a ciascun esempio.

1.1 Decisioni di progetto

Si è scelto di utilizzare un approccio di clustering rigido, nello specifico l'algoritmo K-Means, per segmentare i soggetti in gruppi omogenei sulla base delle loro caratteristiche. Le variabili categoriche presenti nel dataset (come il genere o la lingua madre) sono state codificate mediante one-hot encoding, rendendole compatibili con l'algoritmo. Dopo questa trasformazione, è stato necessario stabilire il numero ottimale di cluster: attraverso il metodo del gomito (elbow method) e l'analisi del Silhou...

Il valore massimo del Silhouette Score è stato ottenuto con K=4 (pari a 0.115), il che suggerisce una discreta separazione tra i cluster. In aggiunta, il WCSS (Within-Cluster Sum of Squares) ha mostrato un buon livello di coesione interna tra i punti di ciascun cluster, indicando che l'algoritmo ha individuato gruppi consistenti.

1.2 K-means

K-means è un algoritmo di clustering che suddivide un insieme di dati in **K cluster**, dove K è un valore predefinito. Inizia posizionando casualmente K centroidi nel dataset e assegna ciascun punto al centroide più vicino. Successivamente, aggiorna la posizione dei centroidi in base alla media dei punti assegnati e ripete il processo fino a convergenza. L'obiettivo è minimizzare la somma dei quadrati delle distanze tra i punti e i centroidi assegnati, creando cluster di dati simili.

Dato che parte delle caratteristiche nel dataset sono di natura categorica e l'algoritmo k-means accetta solo caratteristiche numeriche, è stato necessario convertirle. A tal fine, abbiamo scelto di utilizzare la tecnica di **one-hot encoding**, che crea una nuova variabile binaria univoca per ogni categoria.

Una volta codificate le variabili categoriche, abbiamo dovuto determinare il numero ottimale di cluster. Abbiamo utilizzato la tecnica del **metodo del gomito (elbow method)**, in cui sull'asse delle ordinate sono rappresentati i valori delle somme dei quadrati intra-cluster (**WCSS**), mentre sull'asse delle ascisse sono rappresentati i K cluster. Osservando il grafico del gomito, abbiamo notato che il punto di svolta netto si trova intorno a **K=4**, suggerendo che questo sia il numero ottimale di cluster.

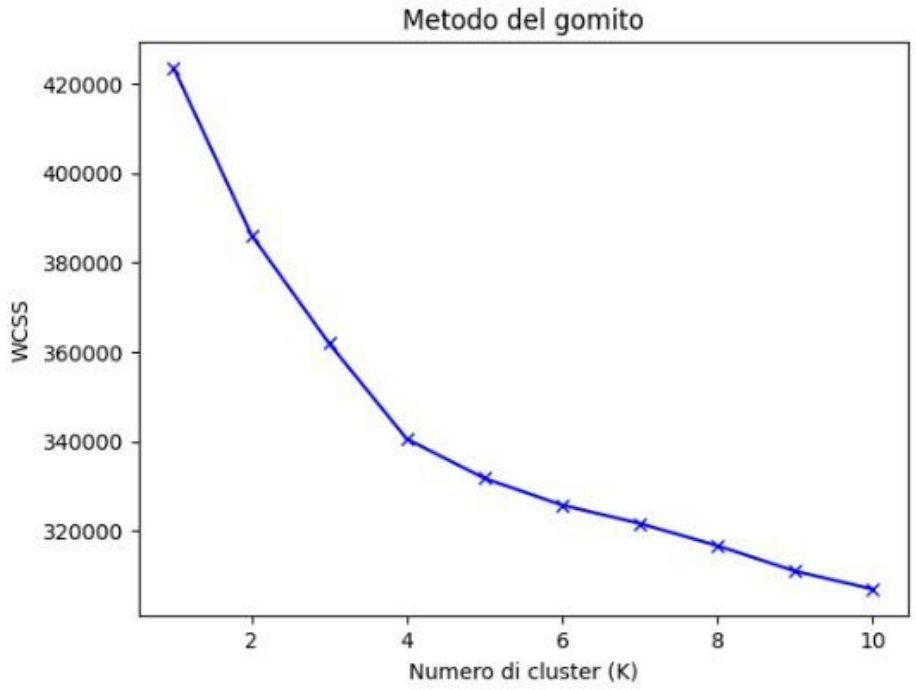
Abbiamo testato diversi valori di **K**:

- Con **K=3**, il **Silhouette Score** ottenuto è stato **0.102**.
- Con **K=4**, il **Silhouette Score** è salito a **0.115**, indicando una migliore separazione tra i cluster.
- Con **K=5**, il **Silhouette Score** è sceso a **0.082**, suggerendo una minore qualità dei cluster.

Poiché **K=4** ha ottenuto il miglior punteggio in termini di separazione dei cluster, è stato scelto come valore ottimale per il clustering.
Le metriche scelte per la configurazione del modello sono state:

- **n_clusters=4**: il numero di cluster in cui il dataset deve essere diviso.
- **random_state=42**: questo parametro controlla la riproducibilità dei risultati. Fissando un valore per random_state, il modello fornirà sempre gli stessi risultati quando viene addestrato con gli stessi dati.
- **n_init=10**: specifica il numero di volte che l'algoritmo viene eseguito con diverse inizializzazioni casuali dei centroidi. L'algoritmo seleziona la soluzione migliore tra i tentativi basandosi sulla somma dei quadrati delle distanze. È stato scelto un valore non troppo alto, ma sufficiente per massimizzare la qualità dell'output.

Grafico:




```
Silhouette Score: 0.115
Clustering completato e dataset salvato.
```

1.3 Valutazione Finale

Sono state valutate le prestazioni del modello utilizzando due metriche:

- **WCSS (Within-Cluster Sum of Squares)**: calcola la somma dei quadrati delle distanze di ogni punto rispetto al proprio centroide di cluster Un valore più basso di WCSS indica che i punti all'interno di ciascun cluster sono più vicini tra loro, suggerendo una maggiore omogeneità dei gruppi individuati..
- **Silhouette Score**: valuta la coesione all'interno dei cluster e la separazione tra i cluster. È calcolato per ciascun punto e rappresenta il rapporto tra la distanza media al proprio cluster e la distanza media ai cluster più vicini. Un punteggio più alto del Silhouette Score indica una migliore separazione dei cluster.

Il valore del Silhouette Score ottenuto è pari a 0.115, il che indica una separazione limitata tra i cluster. Tuttavia, l'analisi dei profili medi ha mostrato che i gruppi individuati presentano differenze significative nelle prestazioni, suggerendo la possibile esistenza di sottotipi di dislessia o soggetti a rischio.

Il dataset finale, contenente una nuova colonna indicante l'assegnazione del cluster per ciascun soggetto, ha permesso di analizzare in modo più strutturato la popolazione studiata. In particolare, si sono potute osservare differenze marcate tra i cluster in termini di prestazioni nei test cognitivi, suggerendo la possibile esistenza di sottotipi di dislessia o di soggetti a rischio, ma non ancora diagnosticati.

Questi risultati possono guidare la comprensione di differenti manifestazioni del disturbo, offrendo una visione più sfumata rispetto alla tradizionale distinzione binaria dislessico/non dislessico.

1.4 Analisi dei profili cluster

Per comprendere meglio le caratteristiche distintive dei gruppi individuati tramite l'analisi di clustering, è stata condotta un'analisi dei profili medi per ciascun cluster. Questa analisi ha permesso di identificare le peculiarità medie dei membri di ogni cluster rispetto a variabili chiave come l'età, i punteggi ottenuti nei test e l'accuratezza delle risposte.

Di seguito, una sintesi delle principali caratteristiche medie per ciascun cluster:

Cluster 0:
- Età media: 12,08 anni
- Punteggio medio nel Test 1: 8,48
- Accuratezza media nel Test 1: 2,27
- Tasso di errore medio nel Test 1: 0,85

Cluster 1:
- Età media: 8,78 anni
- Punteggio medio nel Test 1: 3,01
- Accuratezza media nel Test 1: 3,30
- Tasso di errore medio nel Test 1: 3,08

Cluster 2:
- Età media: 11,76 anni
- Punteggio medio nel Test 1: 0,74
- Accuratezza media nel Test 1: 3,34
- Tasso di errore medio nel Test 1: 10,12

Questi profili medi evidenziano differenze significative tra i cluster. Ad esempio, il Cluster 0 presenta un'età media più elevata e punteggi medi più alti nei test iniziali rispetto al Cluster 1, che è caratterizzato da un'età media inferiore e performance meno soddisfacenti. Il Cluster 2, invece, mostra un'età media simile al Cluster 0, ma con punteggi significativamente più bassi e un tasso di errore più elevato nel Test 1.

L'analisi dettagliata dei profili medi per ciascun cluster è disponibile nel file allegato:

Profili medi cluster

Questa analisi fornisce una comprensione più approfondita delle caratteristiche dei gruppi individuati, facilitando l'interpretazione dei risultati del clustering e supportando eventuali interventi mirati basati sulle specificità di ciascun cluster.

1.5 Integrazione Potenziale nella Classificazione

Sebbene il clustering sia stato condotto in modo indipendente, il dataset clusterizzato costituisce una risorsa preziosa per i modelli supervisionati. L'informazione relativa al cluster di appartenenza può essere utilizzata come feature addizionale nei modelli (Random Forest, SVM, Reti Neurali), migliorando la capacità di generalizzazione. Inoltre, è possibile sviluppare modelli specifici per ciascun cluster, allenandoli in modo mirato sulle caratteristiche peculiari di ciascun sottogruppo.

In alternativa, l'appartenenza ai cluster può essere impiegata per personalizzare il bilanciamento dei dati, per effettuare analisi post-predizione, o per costruire nuove feature ingegnerizzate, derivate dai profili medi dei gruppi.

1.6 Sviluppi futuri

L'introduzione del clustering ha aperto la strada a nuovi scenari di approfondimento. Uno dei principali sviluppi futuri consisterà nell'integrare attivamente i risultati del clustering nei modelli di classificazione, testando empiricamente l'efficacia delle strategie proposte. In particolare, si prevede di confrontare le prestazioni dei modelli tradizionali con quelli che incorporano il cluster come feature o che utilizzano classificatori specializzati per ciascun gruppo.
Un ulteriore ambito di sviluppo riguarda l'utilizzo di tecniche di clustering avanzate, come il clustering morbido (es. Gaussian Mixture Models), che consentirebbero di rappresentare l'appartenenza ai gruppi in maniera probabilistica, riflettendo la natura sfumata dei profili cognitivi dei soggetti.

Infine, si propone di analizzare con strumenti interpretativi (come SHAP o LIME) il peso che l'appartenenza a un cluster esercita nella predizione finale, per migliorare la trasparenza e l'affidabilità clinica del sistema.

In sintesi, il clustering non rappresenta una semplice analisi esplorativa, ma un modulo integrabile, dinamico e altamente informativo, destinato ad arricchire in modo significativo il sistema di supporto alla diagnosi precoce della dislessia.

5. CONCLUSIONI

In questo progetto, abbiamo affrontato il problema della dislessia applicando tecniche di intelligenza artificiale per analizzare e segmentare i dati derivanti da test cognitivi e comportamentali. Lo scopo principale era identificare pattern nascosti e ottenere una comprensione più approfondita della dislessia, con l'obiettivo finale di migliorare il supporto diagnostico.

Attraverso l'applicazione del **clustering rigido (K-Means)**, il dataset è stato suddiviso in quattro cluster significativi. L'analisi ha rivelato che specifiche caratteristiche cognitive e linguistiche influenzano la categorizzazione dei soggetti, suggerendo che esistono gruppi distinti con differenti profili di prestazioni. Sebbene il **Silhouette Score** indichi una separazione non perfetta tra i cluster, i risultati mostrano il potenziale dell'algoritmo nel rilevare strutture latenti nei dati complessi legati ai disturbi dell'apprendimento.

L'analisi dei modelli di apprendimento supervisionato ha evidenziato che:

- La **Neural Network** si è dimostrata la scelta migliore in termini di bilanciamento tra accuratezza e F1-score, offrendo una classificazione robusta ed equilibrata tra le classi.
- **Random Forest** ha raggiunto la massima accuratezza (91%), ma ha mostrato squilibri tra le classi.
- **SVM** ha fornito un buon compromesso tra accuratezza e generalizzazione, mentre **KNN** ha dimostrato prestazioni inferiori, evidenziando difficoltà nel generalizzare i dati complessi.

Questi risultati indicano che i dati sulla dislessia presentano pattern complessi e non lineari, beneficiando quindi di modelli più sofisticati.

6.1 Sviluppi futuri

Il lavoro svolto offre molteplici spunti per futuri sviluppi e miglioramenti:

1. **Clustering morbido:** L'integrazione di tecniche di clustering morbido (soft clustering) potrebbe migliorare l'identificazione delle sovrapposizioni tra gruppi, offrendo una rappresentazione più accurata delle differenze individuali.
2. **Combinazione con modelli supervisionati:** Una direzione promettente sarebbe combinare i risultati del clustering con modelli di apprendimento supervisionato, utilizzando i pattern individuati per migliorare la predizione della dislessia.
3. **Espansione del dataset:** Ampliando il dataset con nuove variabili o integrando dati provenienti da altre fonti (ad esempio, dati genetici, neurofisiologici o ambientali), si potrebbero migliorare ulteriormente le prestazioni e l'accuratezza delle analisi.

4. **Applicazione ad altri disturbi dell’apprendimento:** La metodologia sviluppata potrebbe essere estesa per analizzare altri disturbi dell’apprendimento, come la **discalculia** o il **disturbo da deficit di attenzione e iperattività (ADHD)**, contribuendo alla creazione di nuovi strumenti diagnostici basati sull'intelligenza artificiale.

Questi sviluppi futuri potrebbero portare a sistemi diagnostici più precisi, personalizzati e automatizzati, fornendo un contributo significativo al miglioramento delle diagnosi e del supporto per i disturbi dell'apprendimento.