UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

CASO DI STUDIO
INGEGNERIA DELLA CONOSCENZA
2022-2023

MALARIA VISION

Sistema di Visione Artificiale per la Diagnosi Automatizzata della Malaria tramite Immagini Mediche

Gruppo di Lavoro

Michele Astarita 744455 m.astarita@studenti.uniba.it

Dario Liantonio 704535 d.liantonio3@studenti.uniba.it

Gaetano Angelo Alberto Loizzo 738727 g.loizzo9@studenti.uniba.it

GitHub Repository

Link: https://github.com/GaetanoAA/ICON-22-23.git

Sommario

Introduzione

Il seguente caso di studio, sviluppato tramite l'utilizzo del linguaggio Python, ha come obiettivo la diagnosi automatizzata della malaria basandosi su immagini di cellule reali, reperite dal **National Library of Medicine Official Website.**

Link: https://ceb.nlm.nih.gov/repositories/malaria-datasets/

Il dataset comprende in totale 27558 immagini con istanze uguali di cellule suddivise in due categorie principali: parassitate e non infette.

All'interno di questo studio, sono state effettuate attività di pre-processing e integrazione dei dati per creare una base di conoscenza utilizzando il linguaggio di programmazione Prolog. Sono state ingegnerizzate clausole che consentono di inferire nuove informazioni tramite il ragionamento automatico. Queste clausole sono state poi sfruttate per compiere task di apprendimento supervisionato.

Inoltre, abbiamo affrontato il problema della classificazione delle immagini infette e non infette. Sono stati tentati diversi approcci per gestire le caratteristiche categoriche e numeriche delle immagini. L'obiettivo era trovare una soluzione accurata per avere la massima accuratezza della diagnosi della malaria.

Nel dataset utilizzato sono presenti due tipi di malaria: Plasmodium falciparum e Plasmodium vivax. Di conseguenza, l'obiettivo è quello di ampliare la conoscenza di base per poter permettere a quest'ultima di distinguere, a partire da un'immagine di una cellula, se essa è infetta attraverso approcci alternativi come la dominanza di determinati parametri grafici.

Infine, la conoscenza di base verrà ampliata nuovamente al fine di migliorare i risultati raggiunti nell'argomento di classificazione.

Elenco argomenti di interesse

- **Apprendimento supervisionato:** creato un "Ensemble Model" per analizzare i risultati dei singoli algoritmi di classificazione:
  - Random Forest
  - Decisional Tree
  - Grandient Boosting
  - Ada Boost
  - K-NN
  - Naive Bayes
  - SVC (Support Vector Classifier)

- **Rappresentazione e ragionamento relazionale:** Prolog viene utilizzato per ragionare su una base di conoscenza creata a partire dalle immagini del dataset, consentendo di inferire nuove informazioni.

- **Applicazioni basate su modelli probabilistici relazionali:** sfruttiamo un modello probabilistico precisamente un modello Bayesiano con lo scopo di effettuare una previsione riguardante la parassitosi di una determinata cellula dato uno specifico parametro estratto graficamente.

Preprocessing dei dati

Partendo da questo dataset abbiamo codificato le etichette più importanti per raggiungere un'accuratezza ottimale del progetto. Ovvero:

- **Importazione delle librerie necessarie** per:
    - dividere i dati in un set di addestramento e un set di test
    - creare grafici, nel nostro caso per visualizzare la curva ROC
    - codificare le etichette di classe in valori numerici
    - valutare le prestazioni del modello utilizzando la matrice di confusione e il report di classificazione

In particolare le librerie utilizzate sono: Os, Random, Image.v2, NumPy.

- **Caricamento delle immagini**:
    - specificato il percorso della cartella contenente le immagini da utilizzare.
    - istanziato l'oggetto ImageLoader dal modulo image_loader e vengono caricate le immagini e le etichette.

- **Estrazione delle feature**:
    - istanziato l'oggetto FeatureExtractor dal modulo feature_extractor e vengono estratte le feature dalle immagini caricate.

- **Codifica delle etichette**:
    - istanziato l'oggetto LabelEncoder per convertire le etichette di classe in valori numerici
    - applicato il metodo fit per adattare l'encoder alle etichette e il metodo transform per codificarle

- **Divisione dei dati in set di addestramento e set di test**:
    - utilizzata la funzione train_test_split di sklearn.model_selection per suddividere le feature e le etichette in un set di addestramento e un set di test.

Una volta completata la fase di preprocessing, il codice prosegue con l'addestramento e la valutazione del modello di classificazione, l'utilizzo di un ragionatore Prolog, la valutazione delle prestazioni del classificatore e la visualizzazione dei risultati tramite stampa a console e grafici.

Inserimento feature

Abbiamo preso la decisione di estrarre determinate feature, rappresentate nella tabella sottostante, per rappresentare le informazioni rilevanti contenute nei dati in modo che l'algoritmo possa apprendere i modelli e le relazioni tra le feature e le classi di appartenenza.

NOME	DESCRIZIONE	TIPO
Media e deviazione standard dei canali di colore	Calcolano la media e la deviazione standard dei canali di colore (rosso, verde, blu) per ogni immagine. Queste statistiche possono fornire informazioni sulle caratteristiche cromatiche delle immagini.	intero
Istogramma dei colori nel dominio HSV	convertono l'immagine nel dominio HSV e calcolano l'istogramma dei valori di tonalità (H) utilizzando 8 bin. L'istogramma viene normalizzato per ottenere una distribuzione di probabilità dei colori nel dominio HSV.	intero
Istogramma dei colori nel dominio LAB	convertono l'immagine nel dominio LAB e calcolano l'istogramma dei canali a* e	intero

	b* utilizzando 8 bin. Anche questo istogramma viene normalizzato	
Compressione dell'immagine e colore dominante	riducono le dimensioni dell'immagine a 1x1 pixel e estraggono il colore del pixel risultante come colore dominante	intero

In seguito all'inserimento di queste feature passeremo alla creazione del Knowledge Base.

Creazione Knowledge Base

Abbiamo creato una **base di conoscenza (knowledge base)** nel formato Prolog utilizzando le etichette e le caratteristiche estratte dalle immagini presenti in una specifica cartella. La base di conoscenza contiene informazioni sulle immagini di cellule sanguigne etichettate come "Parasitized" (parassitate) o "Uninfected" (non infette), che rappresentano rispettivamente le cellule sanguigne infettate dal parassita della malaria e le cellule sanguigne normali.

Il processo di creazione del KB avviene attraverso un codice dedicato, presente nel nostro progetto, che permette di riscrivere e aggiornare quest'ultimo in caso di variazione nel dataset; inoltre questo codice si avvale dell'utilizzo di altre classi del progetto per rendere il KB il più completo ed utile ai fini del progetto includendo in esso tutte le features estratte che successivamente verranno utilizzate anche per l'apprendimento supervisionato.

Apprendimento Supervisionato

I vari **algoritmi supervisionati** utilizzati hanno come finalità quella di predire quali cellule presenti nel dataset sono soggette ad infezione da malaria e quali no. Ricordiamo che, gli algoritmi supervisionati sono tecniche di Machine Learning addestrati su un insieme di dati, formato da diverse features ed etichette corrispondenti all'output di riferimento.

Successivamente alla fase di addestramento, avviene quella di predizione che ha come scopo quello di predire, con più accuratezza possibile, una funzione, o come in questo caso, un modello che mappa gli input alle etichette corrispondenti.

Strumenti utilizzati

Di seguito sono descritti brevemente gli algoritmi utilizzati:

1. **Support Vector Machine (SVM)**: Viene utilizzato il classificatore SVC con kernel polinomiale. SVM è un algoritmo di apprendimento supervisionato che può essere utilizzato per la classificazione binaria o multiclasse. Il kernel polinomiale viene utilizzato per mappare i dati in uno spazio dimensionale superiore per separare le classi.

2. **Decision Tree (DT)**: È un algoritmo di apprendimento supervisionato che costruisce un albero di decisione in base alle caratteristiche dei dati di addestramento. Ogni nodo dell'albero rappresenta una caratteristica e una regola di suddivisione che migliora la purezza della classificazione.

3. **Random Forest (RF)**: È un algoritmo di apprendimento di insieme che crea più alberi di decisione utilizzando campioni casuali del set di addestramento. La classificazione finale viene determinata mediante votazione.

4. **Gradient Boosting (GB)**: È un algoritmo di apprendimento di insieme che costruisce gli alberi di decisione in modo sequenziale, mettendo più enfasi sulle istanze che sono state classificate erroneamente dai precedenti alberi. Questo aiuta a migliorare gradualmente le performance del modello.

5. **AdaBoost (AB)**: È un altro algoritmo di apprendimento di insieme che costruisce gli alberi di decisione in modo sequenziale, ma mette più enfasi sugli esempi difficili da classificare.

6. **Naive Bayes (NB)**: È un classificatore probabilistico che si basa sull'applicazione del teorema di Bayes con l'assunzione di indipendenza delle caratteristiche. È particolarmente adatto per dati con molte caratteristiche e può essere utilizzato per problemi di classificazione.

7. **K-Nearest Neighbors (KNN)**: È un algoritmo di classificazione basato sul concetto di "vicinanza". Classifica un'istanza in base alla maggioranza dei voti delle sue K istanze vicine nel set di addestramento.

8. **Multi-Layer Perceptron (MLP)**: È una rete neurale artificiale composta da più strati di neuroni. Può essere utilizzato per problemi di classificazione e apprendimento non lineare.

Libreria associate sono: SkLearn, MathPlotLib.

Decisioni di progetto

La scelta di implementare diversi algoritmi supervisionati è stata effettuata per analizzare quale di essi si addicesse maggiormente al nostro caso di studio e al nostro dataset. Dopo aver effettuato vari test, è risultato che gli algoritmi più compatibili sono: Decisional Tree, Random Foresti, Gradient Boost e Ada Boost. Mentre i meno adatti sono risultati essere: SVM, Naive Bayes e K-NN.

La sostanziale differenza tra gli algoritmi sopracitati è dovuta alle seguenti motivazioni che abbiamo riscontrato.

Nel caso del **K-Nearest Neighbors (KNN)**:

Avendo un dataset di immagini, le diverse feature rappresentano i pixel dell'immagine. Le feature che rappresentano i valori di colore hanno scale molto più ampie rispetto alle feature che rappresentano le coordinate spaziali, la distanza tra i campioni potrebbe essere dominata dalle feature dei valori di colore, influenzando pesantemente la classificazione basata sulla vicinanza dei pixel.
Le immagini sono rappresentate da un alto numero di dimensioni, corrispondenti al numero di pixel dell'immagine. In uno spazio ad **alta dimensionalità**, i campioni tendono ad essere equidistanti, il che può rendere più difficile per KNN trovare vicini significativi e influire sulle prestazioni del classificatore.

```
Algorithm: knn
Training Accuracy: 79.90%
Test Accuracy: 69.32%

Confusion Matrix:
[[1902  871]
 [ 820 1919]]

Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.69      0.69      2773
           1       0.69      0.70      0.69      2739

    accuracy                           0.69      5512
   macro avg       0.69      0.69      0.69      5512
weighted avg       0.69      0.69      0.69      5512
```

Nel caso del **Naive Bayes**:
L'assunzione di indipendenza condizionale delle feature potrebbe essere irrealistica. I pixel di un'immagine possono essere correlati tra loro a causa della struttura spaziale dell'immagine. Questa condizione porta ad una stima errata delle probabilità condizionali basate sulle feature dei pixel e influire sulle prestazioni del classificatore.

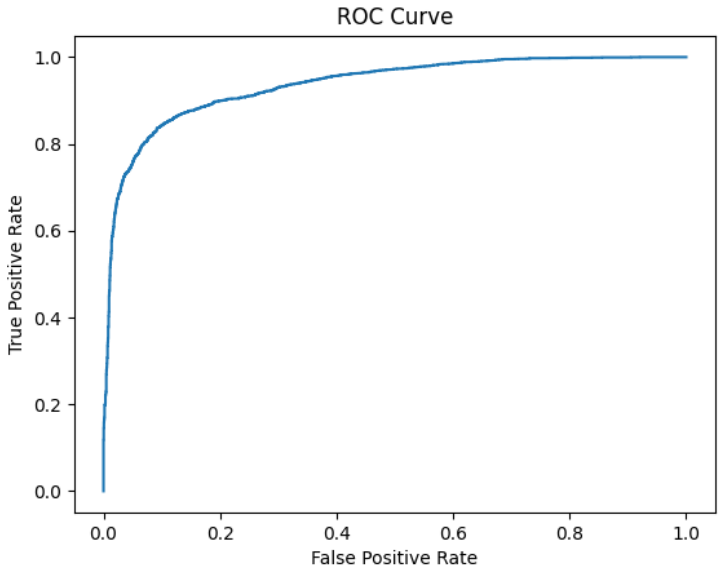
```
Algorithm: nb
Training Accuracy: 66.76%
Test Accuracy: 65.40%

Confusion Matrix:
[[1867  906]
 [1001 1738]]

Classification Report:
              precision    recall  f1-score   support

           0       0.65      0.67      0.66      2773
           1       0.66      0.63      0.65      2739

    accuracy                           0.65      5512
   macro avg       0.65      0.65      0.65      5512
weighted avg       0.65      0.65      0.65      5512
```

Nel caso del Support **Vector Machine (SVM)**:
Le immagini sono soggette a simili condizioni di acquisizione (illuminazione, angolazione, sfondo, etc.). Questa condizione crea difficoltà a trovare un iperpiano di separazione ottimale, portando a una minore accuratezza nel classificare nuove immagini.

```
Algorithm: svm
Training Accuracy: 68.61%
Test Accuracy: 67.51%

Confusion Matrix:
[[1899  874]
 [ 917 1822]]

Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.68      0.68      2773
           1       0.68      0.67      0.67      2739

    accuracy                           0.68      5512
   macro avg       0.68      0.68      0.68      5512
weighted avg       0.68      0.68      0.68      5512
```

Il modello SVM mostra una precisione simile sia per il training che per il test, attorno al 67%. Questo potrebbe indicare che il modello non è in grado di catturare in modo ottimale la complessità del problema e potrebbe essere meno efficace nel distinguere le classi rispetto ad altri algoritmi.

L'uso dei diversi algoritmi è stato combinato attraverso la creazione di un **Ensemble Model**, con la finalità di fornire un'accuratezza media dei vari algoritmi coinvolti e farci un'idea generale su come possa venire classificato il nostro dataset in correlazione alla quasi totalità degli algoritmi supervisionati persi in causa. Al fine di trarre delle conclusioni dettagliate e specifiche, nei vari algoritmi e nell'Ensemble Model sono state implementate varie **valutazioni metriche**, come l'accuratezza, la matrice di confusione e un rapporto di classificazione riepilogativo. Infine, sono state fatte delle valutazioni a livello di sensibilità e specificità dell'Ensemble Model, considerando anche la **curva ROC** e l'**AUC**.



La curva ROC rappresenta graficamente la capacità di discriminazione del modello al variare della soglia di classificazione, in altre parole, indica la percentuale di esempi positivi correttamente classificati rispetto al totale degli esempi positivi presenti.

L'**AUC** indica graficamente l'area sottesa del grafico sotto la curva ROC e misura le capacità di discriminazione del modello. Essa varia tra 0 e 1 con quest'ultimo che indica una capacità di discriminazione perfetta. Sintetizzando il tutto, i due strumenti valutano le performance del modello di classificazione utilizzato.

Un ulteriore decisione progettuale effettuata è stata quella di introdurre ulteriori features, come l'istogramma dei colori nel dominio LAB e HSV poiché, con la loro introduzione, abbiamo notato che su diversi casi di test un incremento dell'accuratezza di ogni algoritmo utilizzato, in particolare l'SVM.

Valutazione

In seguito a un centinaio di test effettuati sul nostro progetto, abbiamo tratto in conclusione che gli algoritmi supervisionati maggiormente compatibili al nostro obbiettivo sono:

Decisional Tree:

```
Algorithm: dt
Training Accuracy: 100.00%
Test Accuracy: 84.09%

Confusion Matrix:
[[2354  419]
 [ 458 2281]]

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.85      0.84      2773
           1       0.84      0.83      0.84      2739

    accuracy                           0.84      5512
   macro avg       0.84      0.84      0.84      5512
weighted avg       0.84      0.84      0.84      5512
```

Il modello Decision Tree mostra una precisione perfetta durante il training (100.00%), ma una precisione leggermente inferiore durante il test (84.58%). Questo suggerisce un possibile overfitting del modello ai dati di training e una minore capacità di generalizzazione ai dati di test.

Random Forest:

```
Algorithm: rf
Training Accuracy: 100.00%
Test Accuracy: 88.32%

Confusion Matrix:
[[2457  316]
 [ 328 2411]]

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.89      0.88      2773
           1       0.88      0.88      0.88      2739

    accuracy                           0.88      5512
   macro avg       0.88      0.88      0.88      5512
weighted avg       0.88      0.88      0.88      5512
```

Il modello Random Forest presenta una precisione perfetta sia per il training che per il test, indicando una buona capacità di adattamento ai dati di training e una buona capacità di generalizzazione ai dati di test.

Gradient Boosting:

```
Algorithm: gb
Training Accuracy: 88.39%
Test Accuracy: 87.28%

Confusion Matrix:
[[2458  315]
 [ 386 2353]]

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.89      0.88      2773
           1       0.88      0.86      0.87      2739

    accuracy                           0.87      5512
   macro avg       0.87      0.87      0.87      5512
weighted avg       0.87      0.87      0.87      5512
```

Il modello Gradient Boosting mostra una precisione elevata sia per il training che per il test, con risultati simili. Ciò indica una buona capacità di generalizzazione del modello ai dati di test.

Ada Boost:

```
Algorithm: ab
Training Accuracy: 86.91%
Test Accuracy: 86.38%

Confusion Matrix:
[[2465  308]
 [ 443 2296]]

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.89      0.87      2773
           1       0.88      0.84      0.86      2739

    accuracy                           0.86      5512
   macro avg       0.86      0.86      0.86      5512
weighted avg       0.86      0.86      0.86      5512
```

Il modello AdaBoost mostra una precisione simile sia per il training che per il test, attestandosi intorno all'86%. Anche in questo caso, i risultati indicano una buona capacità di generalizzazione del modello.

Gli ottimi esiti ottenuti dagli algoritmi sopracitati sono dovuti all'ottima compatibilità del Decision Tree con il nostro caso di studio, in quanto, l'albero di decisone basato su scelte sequenziali, in base ai valori delle features, è il più appropriato per la predizione di cellule infette e non. Le features estratte si addicono particolarmente a questi algoritmi essendo quasi nella loro totalità features numeriche e non categoriche. Ovviamente, Random Forest, Gradient Boosting e Ada Boost ottengono ugualmente ottimi esiti conseguentemente al fatto che anch'essi hanno alla base dei loro algoritmi il Decisional Tree sopracitato.

Rappresentazione e ragionamento relazionale

Abbiamo scelto di utilizzare **Prolog** per effettuare un ragionamento sulla logica di primo ordine. Prolog è strutturato attraverso l'utilizzo di regole logiche come: predicati, che descrivono relazioni tra oggetti, e clausole che definiscono l'inferenza di informazioni dalle relazioni esistenti. Il ragionamento Prolog avviene tramite la risoluzione di query che vengono effettuate a portano all'unificazione di predicati, presenti nel KB a disposizione per ricevere informazioni aggiuntive.

Strumenti utilizzati

SWI-Prolog: ambiente di sviluppo che fornisce un ambiente interattivo in cui è possibile scrivere e testare programmi Prolog traverso file sorgente, esecuzioni di query interattive e visualizzazione dei risultati.

Librerie associate: PySwip, Os, Cv2.

Decisioni di progetto

Nel nostro progetto, abbiamo deciso di convertire il nostro dataset in una base di conoscenza attraverso un codice creato ad hoc per facilitarci il lavoro nel creare probabilità condizionate (nel modello probabilistico di cui discuteremo successivamente) e per sviluppare un approccio differente per l'identificazione di cellule parassitate sulla base di features grafiche come la dominanza di un determinato colore di un'immagine stessa.

Discutendone tra i vari componenti del gruppo e informandoci sul web, oltre che aver analizzato attentamente il dataset, abbiamo notato che le cellule infette rispetto a quelle non infette presentano delle macchie violacee al loro interno.

Di conseguenza a questa valutazione, abbiamo deciso di estrarre come features aggiuntive la dominanza dei colori **rosso, verde e blu (RGB)** per ogni immagine presente nel dataset. La componente violacea abbiamo riscontrato che influisce in modo significativo la dominanza del colore rosso e del colore blu, particolarmente di quest'ultimo; così abbiamo deciso di effettuare un'interrogazione che identificasse cellule infette e non infette sulla base della parametrizzazione di quest'ultimo valore (più di 200 infette).

Valutazione

L'esito delle query effettuate in Prolog ci ha agevolato sulla raccolta di informazioni utili al fine dello sviluppo dell'intero progetto, nel caso della prima query, per la formazione di probabilità condizionate che ci hanno dato la possibilità di prevedere in percentuale la possibilità di predizione di un'immagine parassitata dal dataset rispetto ad una non parassitata.

Come esito della seconda query abbiamo ottenuto una correttezza del 68% nell'identificazione delle varie tipologie di cellule in base alla dominanza del colore blu.

```
Number of Parasitized Images: 13779
Number of Uninfected Images: 13779
```

```
above_threshold(B) :-
    immagine(_, _, L),
    L = [_ | Rest],
    last(Rest, B),
    B > 200.0.

below_threshold(B) :-
    immagine(_, _, L),
    L = [_ | Rest],
    last(Rest, B),
    B < 200.0.
```

```
Immagini con macchie violacee:
C100P61ThinF_IMG_20150918_144104_cell_162
C100P61ThinF_IMG_20150918_144104_cell_163
C100P61ThinF_IMG_20150918_144104_cell_164
C100P61ThinF_IMG_20150918_144104_cell_165
C100P61ThinF_IMG_20150918_144104_cell_166

Immagini senza macchie violacee:
C100P61ThinF_IMG_20150918_145422_cell_169
C101P62ThinF_IMG_20150918_151006_cell_65
C101P62ThinF_IMG_20150918_151006_cell_86
C101P62ThinF_IMG_20150918_151149_cell_71
C101P62ThinF_IMG_20150918_151149_cell_79
```

Applicazioni basate su modelli probabilistici relazionali

Il nostro caso di studio utilizza un modello probabilistico basato su una rete Bayesiana per calcolare le probabilità condizionate sulla variabile "blue" rispetto alle classi "parasitized" e "uninfected" per un insieme di immagini presenti in un determinato dataset. Il contesto relativo è la malaria, dall'analisi di immagini cellulari, per stimare la probabilità che un'immagine cellulare sia infetta (parasitized) o non infetta (uninfected) in base al valore del canale blu dell'immagine. Questo può essere utile per identificare la presenza di parassiti nelle cellule tramite un'analisi delle caratteristiche dell'immagine.

Il modello di rete Bayesiana viene addestrato utilizzando le immagini di per stimare le distribuzioni di probabilità condizionate sulla base del valore del canale blu.

Questo approccio probabilistico fornisce informazioni utili per l'analisi delle immagini cellulari e supportare l'automazione di processi diagnostici o di screening per la malaria o altre condizioni correlate.

Librerie utilizzate: Os, Cv2, Numpy, PIL, PGMPY.

Decisioni di progetto

Parallelamente allo sviluppo del ragionamento Prolog sulla dominanza del colore blu, abbiamo deciso di continuare tale logica di applicazione anche applicato al modello probabilistico per ottenere probabilità condizionate che descrivessero al meglio la distinzione tra cellule infette e non infette attraverso un approccio alternativo, come quello di identificazione in base alla label di appartenenza, ma sfruttando features grafiche come quelle descritto nel paragrafo precedente.

```
    # Scorrere i file all'interno della cartella e delle sottocartelle
    for root, dirs, files in os.walk(folder_path):
        for file_name in files:
            # Verifica se il file è un'immagine (puoi aggiungere ulteriori controlli se necessario)
            if file_name.endswith('.jpg') or file_name.endswith('.png'):
                file_path = os.path.join(root, file_name)
                blue_value = extract_dominant_blue(file_path)  # Estrae il valore blu dominante
                # print(file_path, blue_value)
                if blue_value > 160:
                    count_above += 1
                else:
                    count_below += 1
                total += 1  # Incrementa total per ogni immagine valida

    # Controlla se total è zero per evitare la divisione per zero
    if total == 0:
        probability_above = 0
        probability_below = 0
    else:
        probability_above = count_above / total
        probability_below = count_below / total

    self.cpd_parasitized.values = [[probability_above], [probability_below]]
    self.cpd_uninfected.values = [[1 - probability_above], [1 - probability_below]]
```

Valutazione

L'output sottostante conferma che il processo di distinzione delle cellule in base al parametro di dominanza del colore blu raggiunge esiti quasi del tutto accurati (± 3%), in quanto il dataset è perfettamente bilanciato e, in un sistema ideale, tali probabilità dovrebbero raggiungere, nel nostro caso, una probabilità del 50/50.

```
Probability (Blue > 160): 0.4792074896581755
Probability (Blue < 160): 0.5207925103418245
```

Validazione

Ai fini di rendere il nostro progetto valido e affinché gli output ottenuti possano essere verificati in qualche modo, abbiamo inserito implicitamente delle fasi di validazione all'interno del codice, come:

- Divisione dei dati in train set e test set: Utilizzando la funzione **train_test_split** del modulo **sklearn.model_selection**, i tuoi dati vengono suddivisi in un set di addestramento e un set di test. Questa suddivisione permette di allenare il modello sui dati di addestramento e poi valutarlo sui dati di test per stimare le sue prestazioni.

- Valutazione delle prestazioni del modello: Dopo aver addestrato il tuo modello di ensemble, utilizzi la classe **ClassifierEvaluator** per valutare le prestazioni del classificatore. Calcoli l'AUC della curva ROC e i tassi di falsi positivi e veri positivi. Queste metriche sono spesso utilizzate come misure di valutazione per i modelli di classificazione.

Conclusioni

In conclusione, il progetto presentato offre una soluzione per la diagnosi della malaria mediante l'analisi di immagini di cellule infette e non infette. Combina diverse tecniche e strumenti, tra cui estrazione di feature, modelli di ensemble, reasoning con Prolog e l'utilizzo di una rete bayesiana.

Le possibili applicazioni di questo progetto nell'ambito medico sono molteplici. Il sistema sviluppato può essere utilizzato per automatizzare il processo di diagnosi della malaria, fornendo una valutazione rapida e precisa delle immagini delle cellule. Ciò può consentire di accelerare il processo diagnostico e migliorare l'efficienza. Il sistema può essere utilizzato come strumento di supporto decisionale per i medici. Fornisce informazioni dettagliate sui risultati del classificatore e dei modelli di ensemble, inclusi i punteggi di accuratezza, le matrici di confusione e i report di classificazione. Queste informazioni possono aiutare i medici a prendere decisioni informate e migliorare la precisione della diagnosi.

Il progetto offre anche la possibilità di eseguire analisi approfondite sui dati raccolti. Ad esempio, è possibile esaminare le caratteristiche estratte dalle immagini per identificare correlazioni o pattern specifici associati alla malaria. Queste informazioni possono contribuire alla ricerca medica0.

In sintesi, il progetto rappresenta un'applicazione pratica e innovativa delle tecnologie di intelligenza artificiale e dell'analisi delle immagini nel campo della diagnosi della malaria. Offre un approccio automatizzato, accurato e tempestivo alla diagnosi, migliorando l'efficienza e l'efficacia del processo diagnostico e potenzialmente contribuendo a ridurre l'impatto della malattia.