Classificazione delle Condizioni Tiroidee mediante Machine Learning e Deep Learning[1]

Esplorazione del Dataset

Il dataset[2] analizzato contiene dati clinici e sierologici di pazienti per la classificazione di condizioni tiroidee. L'obiettivo principale è stato quello di sviluppare modelli predittivi per distinguere tra tre classi di diagnosi: condizioni normali, ipertiroidismo e ipotiroidismo. Dopo aver selezionato le diagnosi precise riguardanti le condizioni tiroidee e mappato le condizioni nelle tre classi principali, è emerso un significativo sbilanciamento: Condizioni normali: 89.73%, Ipotiroidismo: 7.86%, Ipertiroidismo: 2.41%

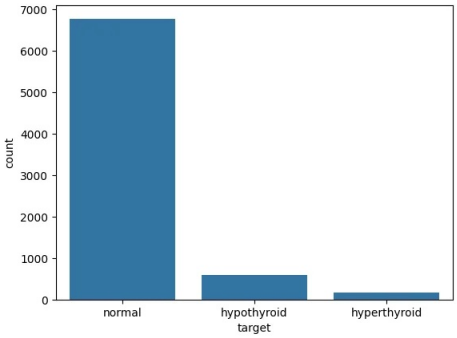


Figura 1: Distribuzione delle classi target

L'analisi esplorativa ha rivelato che i valori medi di TT4 differiscono significativamente tra le tre classi, suggerendo il potenziale discriminativo di questa variabile (vedi notebook). La matrice di correlazione dei valori sierologici ha evidenziato una forte correlazione tra TT4 e FTI, come atteso dato che FTI è calcolato mediante una formula approssimativa basata su TT4.

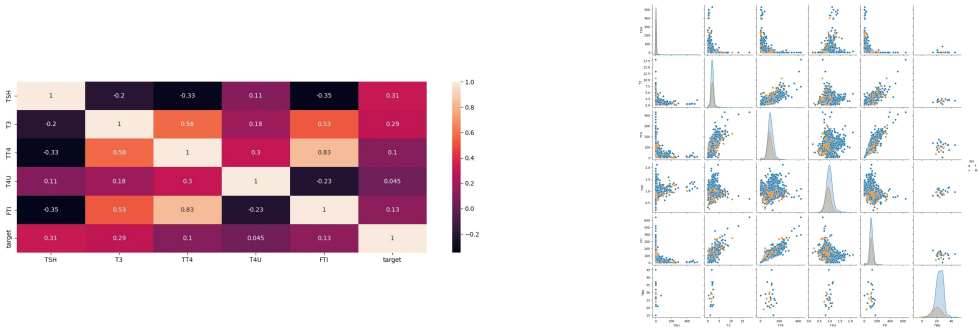


Figura 2: Matrice di correlazione dei valori sierologici

La distribuzione dei dati per sesso mostra omogeneità nei valori clinici, sebbene vi sia una maggiore frequenza di pazienti di sesso femminile nel dataset.

Cleaning e Preprocessing

Il processo di pulizia ha comportato la rimozione delle colonne non informative: TSH_measured, T3_measured, TT4_measured, T4U_measured, FTI_measured, TBG_measured, referral_source, patient_id, query_hypothyroid, query_hyperthyroid, query_on_thyroxine. Le colonne 'TBG' e 'hypopituitary' sono state

[1]Autrice: Dott.ssa Teresa Staropoli

[2]https://github.com/therestar/thyroid-classification-model

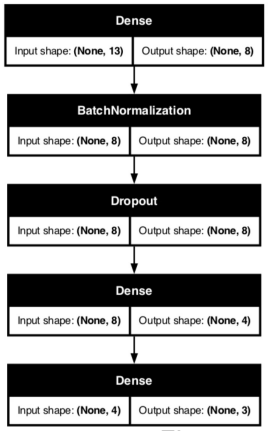
rimosse per l'eccessiva presenza di valori nulli, anche i record con età superiore a 100 anni (valori inconsistenti). Sono state rimosse le istanze con più di 2 valori sierologici mancanti e quelle senza indicazione del sesso. Il dataset pulito finale contiene 6933 righe e 18 variabili, inclusa la variabile 'target'. I dati sono stati suddivisi con stratificazione sulla variabile 'target': Test set (15% dei dati), Training set (70% dei dati), Validation set (15% dei dati). Il preprocessing ha incluso l'imputazione dei valori sierologici mancanti e la normalizzazione di questi valori. Tramite Sequential Feature Selection (SFS) sono state selezionate 13 features ottimali, riducendo le dimensioni da 17 a 13 variabili. Le dimensioni finali sono: Training (4852, 13), Validation (1041, 13), Test (1040, 13).

Creazione Modelli Machine Learning e Deep Learning

È stato implementato un modello Random Forest con ottimizzazione degli iperparametri tramite RandomSearchCV. I parametri ottimali identificati sono: n_estimators: 200, max_depth: 30, min_samples_split: 10, min_samples_leaf: 4, criterion: gini, class_weight: balanced (per gestire lo sbilanciamento delle classi)

Figura 3: Configurazione del modello Random Forest

L'architettura della rete neurale implementata con Keras comprende: Layer di input (13 features), Primo hidden layer con 8 neuroni con attivazione ReLU, Batch Normalization e Dropout (rate=0.1). Il secondo hidden layer ha 4 neuroni con attivazione ReLU, infine l'output layer ha 3 neuroni con attivazione Softmax perché l'obiettivo è una classificazione multiclasse.



Layer (type)	Output Shape	Param #
dense (Dense)	(None, 8)	112
batch_normalization (BatchNormalization)	(None, 8)	32
dropout (Dropout)	(None, 8)	0
dense_1 (Dense)	(None, 4)	36
dense_2 (Dense)	(None, 3)	15

Total params: 195 (780.00 B)

Trainable params: 179 (716.00 B)

Non-trainable params: 16 (64.00 B)

Figura 4: Architettura del modello Deep Learning

Il modello utilizza l'ottimizzatore Adam con learning rate adattivo, la loss function 'Sparse Categorical Crossentropy' e callbacks come 'EarlyStopping' e 'ReduceLROnPlateau' per ottimizzare gli iperparametri. 'Batch Normalization' e 'Dropout' sono stati implementati per ridurre l'overfitting.

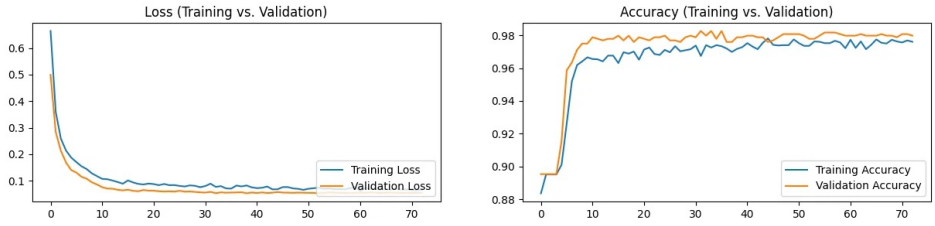
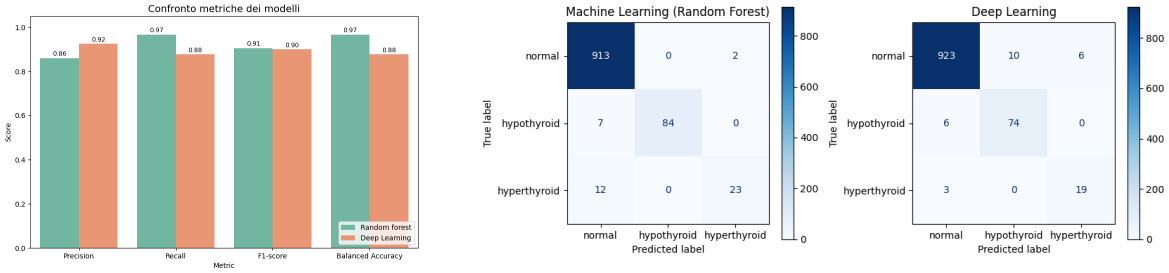


Figura 5: Curve di training - Loss e Accuracy

Benchmarks

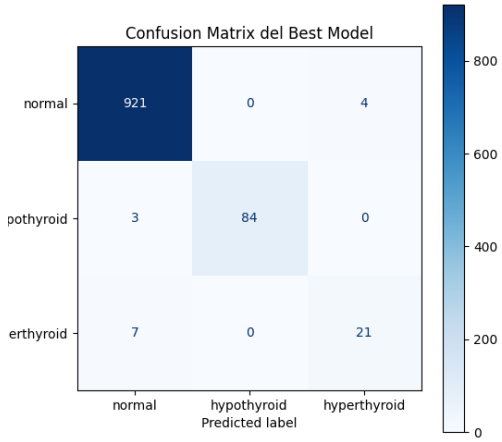
Per confrontare i modelli sono state utilizzate le metriche dei classification reports (precision, recall, F1-score) e la balanced accuracy per gestire lo sbilanciamento delle classi. Sono state visualizzate le matrici di confusione per analizzare gli errori di classificazione e confrontate le balanced accuracies dei due modelli tramite barplot. Le performance sul validation set mostrano che entrambi i modelli classificano accuratamente la classe maggioritaria (condizioni normali) e beneficiano del bilanciamento implementato per le classi



minoritarie.

Figura 6: Confronto tra metriche dei modelli: Report di classificazione e Matrice di confusione.

Il modello migliore è stato selezionato in base all'accuratezza bilanciata sul validation set, infine è stato



valutato sui dati di test, mai visti durante il training. La balanced accuracy finale (0.9431) conferma la capacità di generalizzazione del modello e la sua efficacia nella classificazione delle condizioni tiroidee.

Figura 7: Matrice di confusione del modello Random Forest sul test set.