UNIVERSITÀ
DI PARMA

DIPARTIMENTO DI SCIENZE
MATEMATICHE, FISICHE
E INFORMATICHE

Corso di Laurea in Informatica

Machine Learning per l'analisi predittiva dei tumori testa collo

Machine learning for predictive analysis
of head and neck tumors

Relatore:
Prof. Roberto Alfieri

Correlatore:
Dott. Massimiliano Turchetto

Laureanda:
Marta Caffagnini

Anno accademico 2020/2021

Indice

Abstract

Il Carcinoma a Cellule Squamose del Cavo Orale OSCC è uno dei Tumori Testa-Collo peggio stratificati, in particolare quello non correlato all'infezione da HPV. Nei pazienti affetti da OSCC la Sopravvivenza Globale è del 60% ma varia moltissimo – dal 10% all'80% - in relazione alle dimensioni del tumore e alle sue caratteristiche biologiche.

Pertanto è necessario, al momento della diagnosi, riuscire a differenziare i tumori con un comportamento biologico più benigno da quelli più aggressivi.

Tuttavia, ad oggi il sistema di stadiazione utilizzato nella pratica clinica – il sistema TNM – non è in grado di differenziare questi due diversi gruppi di tumori. Tale limitazione è principalmente dovuta al fatto che il sistema di stadiazione in questione non prende in considerazione né le caratteristiche individuali del paziente né la biologia del tumore e le sue caratteristiche genomiche.

All'interno di nuovi modelli prognostici che utilizzano la Rete Neurale e Random Forest sono stati inseriti numerosi dati riguardanti sia il paziente, tra cui età, sesso, fattori di rischio, sia il tumore stesso come ad esempio l'aspetto anatomopatologico, micro e macroscopico, e le caratteristiche radiologiche e genomiche.

L'analisi di questa grande quantità di dati, che sarebbe stata impossibile da svolgere con i classici metodi statistici, è stata processata attraverso Big Data, un particolare tipo di Intelligenza Artificiale basato sul Machine Learning al fine di trovare nuove correlazioni tra i dati stessi.

Tramite l'applicazione di Random Forest, siamo stati in grado di selezionare all'interno del gruppo di dati immessi come input nel modello, quelli che risultavano essere più caratterizzanti i tumori dei nostri pazienti.

In particolare, è stata effettuata una selezione dei dati genomici più rilevanti per la prognosi dei pazienti, in base ai modelli di Machine Learning. Successivamente li

abbiamo descritti ed analizzati in modo da poter meglio caratterizzare il profilo biologico e molecolare del tumore.

Una migliore stratificazione e classificazione dei tumori dei pazienti aiuterebbe a prendere decisioni terapeutiche più adeguate, evitando il trattamento eccessivo, che è spesso associato ad un peggioramento della qualità di vita del paziente, ed in generale ad aprire le porte verso un nuovo concetto di medicina personalizzata.

1. Introduzione

In questo capitolo è spiegato l'obiettivo della tesi e come il machine learning si inserisce in questo studio. Sono fornite inoltre le informazioni mediche necessarie alla comprensione dei capitoli successivi. Una sezione è dedicata ai tumori testa-collo e alle criticità che si presentano nella scelta del trattamento guidata dal TNM.

1.1 Machine learning e analisi predittiva dei tumori

L'Azienda Ospedaliera di Parma (AOP) partecipa al progetto Europeo BD2Decide ("Big Data and models for personalized Head and Neck Cancer Decision support") che nasce con l'obiettivo di realizzare una migliore stratificazione dei tumori testa-collo (HNC). Con stratificazione si intende un processo statistico che classifica unità (in questo caso tumori) che hanno caratteri comuni. I tumori testa-collo hanno un'elevata incidenza e mortalità, infatti sono la sesta causa di morte per neoplasia a livello mondiale e in Europa la loro incidenza ammonta a 150.000 nuovi casi all'anno, con una mortalità pari a 70.000 casi all'anno.

La scelta della terapia, guidata da una migliore stratificazione, ne gioverebbe. Il trattamento applicato a questi tumori è spesso troppo aggressivo abbassando drasticamente la qualità della vita del paziente che faticherà a deglutire, respirare o parlare.

La stadiazione è un momento molto importante nella diagnosi di un tumore, perché permette di formulare una prognosi e di scegliere il tipo di trattamento più adatto al paziente.

Il sistema di stadiazione attuale è il TNM, un protocollo in evoluzione, che si vuole rendere più preciso nella scelta di trattamenti adeguati al paziente in questione.

Il TNM non considera le caratteristiche specifiche del paziente e le caratteristiche biologiche del tumore.

L'obiettivo è trovare un metodo di classificazione più completo che comprende tecnologie basate sull'Intelligenza Artificiale.

In particolare il progetto BD2Decide ha l'obiettivo di creare modelli di supporto alla decisione per i tumori testa-collo.

Migliorando la precisione dell'attuale predizione della prognosi nel tumore di testa e collo, il sistema ambisce a realizzare un trattamento massimizzando i risultati terapeutici e minimizzando l'impatto della terapia.

Dal punto di vista computazionale viene impiegato il machine learning (ML), una branca dell'Intelligenza Artificiale.

Il ML è un metodo di analisi dati che automatizza la costruzione di modelli analitici. Questi sistemi possono imparare dai dati, identificano modelli autonomamente e prendono decisioni con un intervento umano minimo.

I modelli di ML, elaborando molti dati di esempio per un task, trovano statisticamente dei pattern che poi vengono utilizzati come regole per automatizzare il lavoro.

Nel caso dell'apprendimento supervisionato i modelli estraggono rappresentazioni dai dati forniti. Le rappresentazioni sono modi differenti di vedere e codificare i dati. Questi devono comprendere:

- dati di input che possono essere immagini, file audio

- esempi di output attesi

- parametri per monitorare se l'algoritmo fa un buon lavoro. Viene misurata la lontananza tra l'output calcolato e quello corretto in modo da migliorare l'algoritmo. L'aggiornamento dell'algoritmo è chiamato apprendimento.

Ad esempio, per automatizzare il riconoscimento delle foto delle vacanze, bisogna sottoporre al sistema di machine learning molti esempi di immagini già classificate dagli umani. Il sistema creerà dei pattern usati come regole per associare le foto ai rispettivi tag.

Gli algoritmi di ML cercano automaticamente le trasformazioni che, applicate ai dati di input, creano rappresentazioni più utili per un certo compito.

Il Deep Learning è una branca del ML, in cui l'apprendimento consiste in più strati che creano rappresentazioni sempre più significative.

L'aggettivo "deep" è dovuto ai vari strati di rappresentazione che caratterizzano l'apprendimento.

I modelli di deep learning organizzati a strati sono chiamati *reti neurali*.

Ogni strato esegue delle trasformazioni sui dati di input. Queste trasformazioni sono parametrizzate in funzione dei pesi che sono associati a ogni strato.

1.2 Tumori testa-collo

Carcinoma a cellule squamose testa e collo (HNSCC) è un tumore che nasce dalle cellule squamose nel rivestimento della mucosa della cavità orale, dell'orofaringe, ipofaringe e laringe. Rappresenta più del 90% di tutti i tumori della testa e del collo.

Ci sono 5 tipi principali di HNC:

- tumore laringeo e ipofaringeo

- tumore della cavità nasale e paranasale (nasal cavity and paranasal sinus cancer)

- tumore nasofaringeo

- tumore orale e orofaringeo (OSCC)

- tumore della ghiandola salivare

Epidemiologia

HNSCC è il settimo tumore più comune nel mondo e il sesto per mortalità.
Circa 630,000 nuovi casi sono diagnosticati ogni anno, di cui circa 150,000 in Europa.

HNSCC è più comune negli uomini tra i cinquanta e i sessant'anni, ma l'incidenza nei più giovani sta aumentando a causa dell'infezione da papillomavirus (HPV).
In Italia l'incidenza è di 9.6 casi su 100,000 per gli uomini e 3.1 casi su 100,000 per le donne.

Fattori di rischio

HNSCC è causato da vari fattori che possono alterare il DNA delle cellule e causare il tumore. I fattori di maggiore rischio sono il consumo di alcol e tabacco, l'infezione da HPV e l'inquinamento. Altri rischi sono l'invecchiamento, scarsa igiene orale, diete povere di verdure. Anche fattori genetici contribuiscono al rischio di HNSCC.

Caratteristiche cliniche

HNSCC è spesso asintomatico nei primi stadi, infatti solo 1 tumore su 3 è diagnosticato nei primi stadi. Più lo stadio aumenta più i trattamenti necessari sono aggressivi e più probabili sono le complicazioni.

Classificazione, prognosi, trattamento

HNSCC è caratterizzato da alti tassi di ricadute e di mortalità. Dopo 5 anni dalla prima diagnosi, solo il 60% sopravvive al carcinoma a cellule squamose della cavità orale (OSCC). Questo tumore è solitamente trattato con un intervento chirurgico ed eventualmente con radioterapia o chemioterapia post-intervento. La decisione della modalità di trattamento è basata sul sistema di stadiazione TNM (Sezione 1.3).
L'operazione chirurgica o in alternativa il solo trattamento radioterapico sono sufficienti nei primi stadi, mentre chemioterapia e radioterapia sono spesso necessari nel caso di stadi avanzati. Il TNM è limitato riguardo la personalizzazione del trattamento.

1.3 TNM

Il sistema di stadiazione dei tumori testa-collo è il TNM (Tumour, Node and Metastasis), una classificazione fondata dal AJCC (American Joint Committee on Cancer) per categorizzare i pazienti malati di tumore sulla base delle diverse prognosi.

A ogni tumore viene associato un codice alfanumerico creato in base alle caratteristiche anatomopatologiche del tumore. La T che sta per tumore può essere compresa tra 0 e 4, il valore di N (linfonodo) è compreso tra 0 e 3, mentre M (Metastasi) può essere o 0 o 1. Ai valori di questi parametri viene associato uno stadio compreso tra I e IV.

Il TNM è in continua evoluzione e l'ottava e ultima versione ha introdotto due indicatori prognostici: la profondità dell'invasione (DOI) e l'estensione extra-nodale (ENE).

Profondità dell'invasione (DOI)

Il parametro T è stato migliorato aggiungendo DOI alle caratteristiche del tumore da analizzare oltre alla grandezza della superficie e allo spessore.

Un DOI maggiore di 10 mm è un fattore prognostico significativo e negativo. Questo parametro aiuta a suddividere i pazienti in base alla sopravvivenza.

Estensione extra-nodale (ENE)

Il parametro dei linfonodi (N) è arricchito dall'introduzione dell'estensione extra nodale (ENE). La presenza di estensione extra nodale è definita come estensione del tumore maligno nelle cellule del linfonodo interessato.
A seconda della gravità, può essere riconosciuto clinicamente, con radiografia o con un esame istologico, cioè della struttura microscopica degli organi e dei tessuti.

Stadi dei tumori

Come mostra la Figura 1.1, lo stadio 0 include tumori senza interessamento linfonodale (N0) e senza metastasi (M0), stadio I con tumori di estensione T1 con N0 e M0.
Sono di stadio II i tumori con estensione T2 e N0 e M0.
Lo stadio III rappresenta i tumori con estensione T3, N0, M0 oppure con T1 o T2 o T3 e N1, M0.
Lo stadio IV è suddiviso in 3 sottostadi: IVA, IVB, IVC.
Lo stadio IVA comprende tumori con T4a e N0 o N1 e tumori da T1 a T4 con N2.
Lo stadio IVB rappresenta i tumori T4b con qualsiasi N e i tumori N3.
Tutti i tumori con M1 solo dello stadio IVC.

Categoria T	Categoria N	Categoria M	Stadio
Tis	N0	M0	0
T1	N0	M0	I
T2	N0	M0	II
T3	N0	M0	III
T1, T2, T3	N1	M0	III
T4a	N0, N1	M0	IVA
T1, T2, T3, T4a	N2	M0	IVA
T qualsiasi	N3	M0	IVB
T4b	N qualsiasi	M0	IVB
T qualsiasi	N qualsiasi	M1	IVC

Figura 1.1: TNM dell'ottava edizione

La stratificazione del rischio è comunque approssimativa soprattutto negli stadi avanzati del tumore. Infatti, il TNM raggruppa tumori con parametri di valore diverso in una singola classe di prognostica.

Il sistema di stadiazione TNM valuta solo gli aspetti morfologici del tumore e non quelli funzionali. Si è dimostrato necessario integrare un sistema di stadiazione morfologico e funzionale soprattutto per i tumori a stadio avanzato.

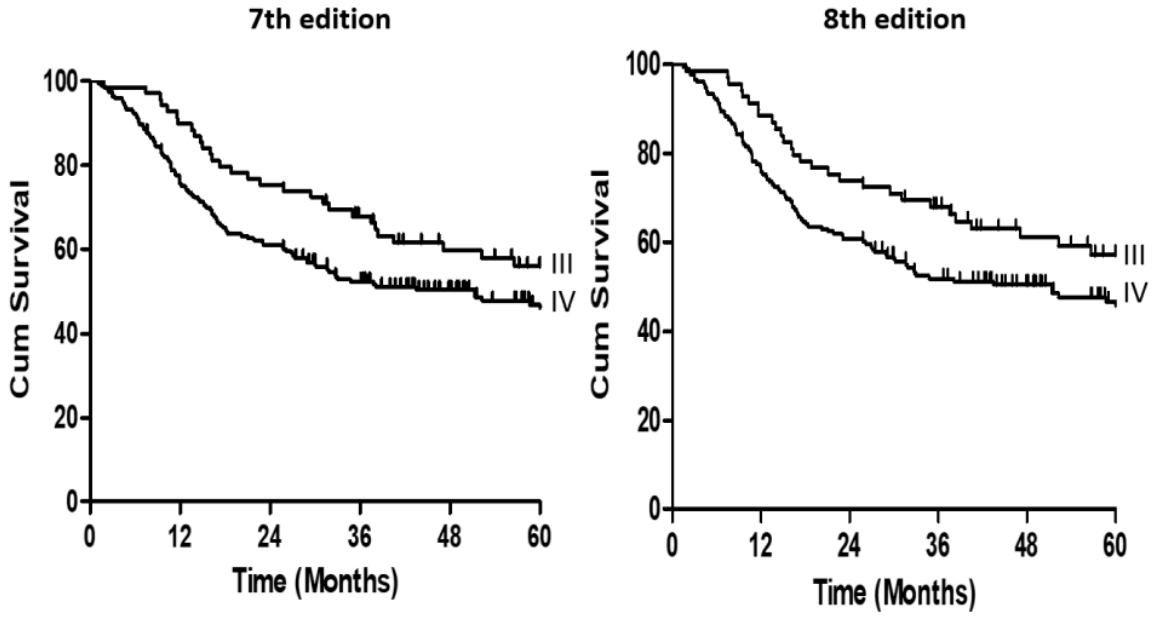

Figura 1.2: Differenza nella sopravvivenza dei pazienti con tumore a stadio avanzato usando il TNM della settima e ottava edizione.

La differenza del tasso di sopravvivenza tra TNM della settima e ottava edizione non è così evidente, come si può vedere dalla Figura 1.2.

La firma genetica

I tumori testa-collo sono molto eterogenei dal punto di vista anatomico e genomico, per la variabilità del comportamento biologico e della risposta alla terapia.
I modelli di prognostica spesso considerano solo le caratteristiche cliniche del tumore, ma il profilo molecolare e genomico del tumore sono necessari per la scelta del trattamento più adeguato per il paziente. Infatti, tumori considerati dello stesso stadio sono molto diversi tra loro dal punto di vista molecolare.

Grazie all'analisi genomica i ricercatori hanno osservato che i geni influenzano il comportamento e la risposta alla terapia dei tumori testa-collo.

La firma genetica è la combinazione di geni selezionati che influenzano il comportamento biologico. Con la firma si può predire la ricaduta locale o nella regione circostante o metastasi distanti.
Per esempio se l'analisi molecolare mostra un alto rischio di metastasi, già nella fase

preoperatoria si può considerare una chemioterapia coadiuvante. Invece, se c'è un alto rischio di ricaduta locale sarà necessario intensificare la radioterapia.

Soprattutto gli stadi più avanzati sono i peggio stratificati e i più frequenti perché il tumore è asintomatico e continua a crescere finché il paziente non se ne accorge.

La stratificazione nelle diverse categorie promossa dal progetto BD2Decide si basa su:

- specifici biomarcatori come l'infezione da papillomavirus (HPV), l'imaging, dati genomici e radiomici.

- epidemiologia legata alla popolazione e fattori di rischio in relazione alle diverse prognosi.

Gli obiettivi sono:

- predire una più accurata situazione clinica del singolo paziente al momento della diagnosi

- per ottenere firme prognostiche specifiche relative a diversi sottotipi HNC e per ogni paziente

- dare la priorità ai fattori prognostici che sono molto significativi per la prognosi e utilizzarli come guida per la scelta del trattamento più appropriato.

- per superare la soggettività delle scelte cliniche.

I dati raccolti da BD2Decide sono troppi e molto eterogenei per essere analizzati con un'analisi statistica tradizionale, per questo è impiegato il machine learning. Questo è inoltre impiegato per scoprire e validare nuovi pattern di prognostica.

2. Materiali e metodi

Questo capitolo presenta i materiali (dati utilizzati e la loro struttura), i metodi (gli algoritmi e i modelli di machine learning come reti neurali, Random Forest, Decision Tree) e gli strumenti offerti da Python per l'analisi dei dati.

I dati e gli strumenti spiegati in questo capitolo saranno impiegati nell'analisi effettuata nel capitolo successivo.

2.1 Dati raccolti

Il progetto BD2Decide ha raccolto i dati di circa 1500 pazienti affetti da HNC in stadio avanzato, coinvolgendo diversi centri in Italia, Germania e altri paesi europei.

Sono stati reclutati circa 1500 pazienti con stadi avanzati (III/IV stadio) del HNSCC.
I pazienti derivano da due diverse sorgenti:

- Pazienti retrospettivi che sono circa 1000 persone curate tra il 2009 e il 2013 provenienti da un precedente progetto.

- Pazienti prospettici che sono circa 450 persone curate più recentemente e raccolte direttamente dal progetto BD2Decide.

Tutti i pazienti selezionati, dopo essere stati informati sullo studio, hanno firmato un consenso informato per partecipare. La popolazione selezionata comprende uomini e donne con un'età compresa tra i 20 e gli 89 anni.

Nella Figura 2.1 è rappresentata la popolazione selezionata.

Patient Characteristics	Retrospective Study		Prospective Study*	
	No.	%	No.	%
Number	**1086**		**389**	
Male/Female	786 /300	72/28	267/121	69/31
Age [median age (range)]	62 (20-93)		60 (20-93)	
Site	n= 1086		n=389*	
Oral cavity	273	25%	126	32%
Oropharynx	443	41%	175	45%
Hypopharynx	130	12%	36	10%
Larynx	240	22%	52	13%
Stage				
III	297	27%	82	21%
IVa	678	63%	277	71%
IVb	111	10%	30	8%
Median follow-up [months]	43		20	

Figura 2.1. La popolazione selezionata nel progetto BD2Decide.

Partendo dai dati dell'AOP dei pazienti del Dipartimento di Maxillo Facciale raccolti per il progetto BD2Decide, è stata fatta un'analisi alternativa.

I dati raccolti dall'AOP e suddivisi in diverse tabelle sono di tipo clinico, radiomico e genomico.

I dati clinici comprendono:

- informazioni sul paziente: la data della prima diagnosi, età e sesso del paziente, fattori di rischio (consumo di alcol e fumo e familiarità), data del decesso se avvenuto.

- informazioni sul HNC: la data della prima diagnosi, il tipo di tumore, l'organo colpito e la posizione, l'estensione del tumore, TNM, complicazioni, recidiva.

- informazioni sul trattamento

- informazioni sulla qualità della vita del paziente prima e dopo il trattamento (dopo 6 e 18 mesi). La qualità di vita del paziente è definita in base a quanto la malattia ha colpito la deglutizione, il linguaggio e alla salute generale del paziente.

Come si vede in Figura 2.1, 92 è il numero di pazienti di cui abbiamo i dati clinici.
La Figura 2.2 mostra che nella tabella ci sono 479 colonne, cioè 479 parametri clinici per ogni paziente.

I dati genomici codificano l'espressione genica del paziente e sono suddivisi in due tabelle: una con i pazienti malati di tumore all'orofaringe e l'altra con i tumori alla cavità orale. Come mostra la Figura 2.1, la tabella dei tumori all'orofaringe comprende i dati di 11 pazienti, mentre quella della cavità orale 136. I dati genomici di entrambe le tabelle sono costituiti da 26904 colonne, come si può vedere dalla Figura 2.2.

I dati radiomici sono le immagini mediche dei pazienti. Ci sono due tabelle, una per le MRI e l'altra con i dati delle immagini CT.

MRI (Magnetic Resonance Imaging) è una tecnica di generazione di immagini usata per scopi diagnostici, basata sul principio fisico della risonanza magnetica nucleare.

Le immagini CT (Computed Tomography) riproducono in sezione l'anatomia e sono create da un'analisi generata al computer, dell'attenuazione di un fascio di raggi X mentre passa attraverso una sezione corporea. Questa tecnica è generalmente nota come TAC.

Come mostra la Figura 2.2, la tabella delle immagini MRI comprende i dati di 224 pazienti, mentre quella delle CT 69 pazienti. I dati radiomici di entrambe le tabelle sono costituiti da 1024 feature (cioè colonne), come si può vedere dalla Figura 2.3.
Le feature sono caratteristiche o parametri di input.

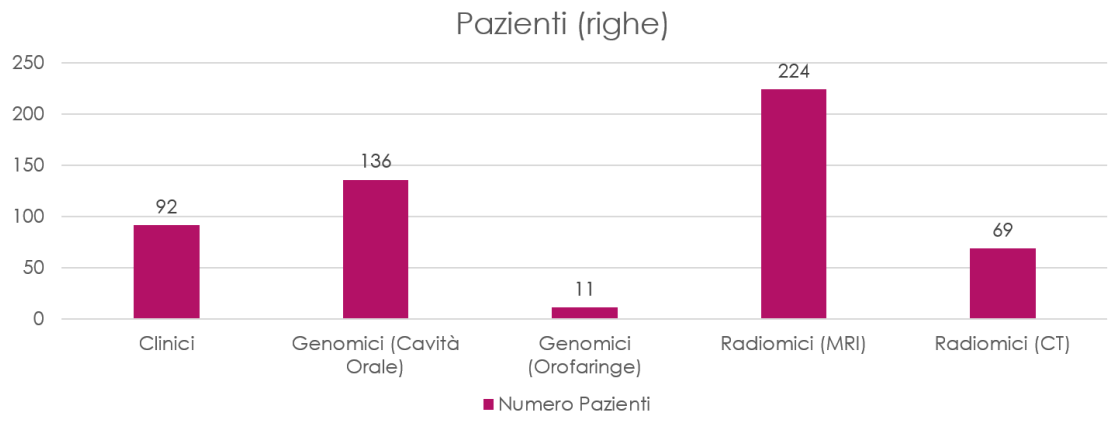

Figura 2.2 Numero di pazienti per ogni tabella

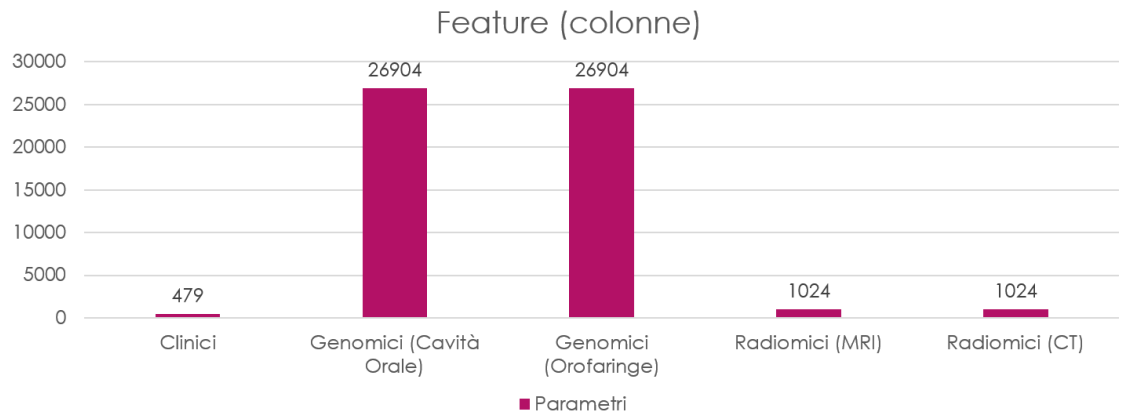

Figura 2.3 Numero di colonne per ogni tabella

Ogni tabella ha circa 100 pazienti e per ogni paziente l'AOP ha raccolto molti dati.
Si tratta di Big Data perché per ogni paziente sono fornite molte informazioni e quelle utilizzate sono anche meno rispetto a quelle originarie che sono state scremate e ridotte per avere dati il più significativo possibile.

2.2 Strumenti di analisi dei dati

I dati grezzi forniti dall'AOP devono essere processati per facilitare il lavoro degli algoritmi di machine learning e per renderli manipolabili.
Le reti neurali non sono in grado di processare testi quindi tutti i dati clinici che sono testuali devono essere codificati e tradotti in valori. A supporto di questi lavori di manipolazione si possono utilizzare librerie Python come NumPy e Pandas. Per la

costruzione di modelli di machine learning Python fornisce Tensorflow, Keras e Scikit-learn.

2.2.1 Pyhton

Python è un linguaggio di programmazione di più "alto livello" rispetto alla maggior parte degli altri linguaggi, è orientato agli oggetti e adatto, tra gli altri usi, a sviluppare applicazioni distribuite, scripting, computazione numerica e system testing.

Spesso viene anche studiato tra i primi linguaggi per la sua somiglianza a uno pseudo-codice e di frequente viene usato per simulare la creazione di software grazie alla flessibilità di sperimentazione consentita da Python.

È un linguaggio multi-paradigma che ha tra i principali obiettivi: dinamicità, semplicità e flessibilità. Supporta il paradigma object oriented e la programmazione strutturata.

Le caratteristiche più evidenti sono le variabili non tipizzate e l'uso dell'indentazione per la sintassi delle specifiche al posto delle parentesi.

Altre caratteristiche distintive sono l'overloading di operatori e funzioni tramite delegati, la presenza di un ricco assortimento di tipi e funzioni di base e librerie standard, sintassi avanzate quali slicing e list comprehension.

Il controllo dei tipi è forte (strong typing) e viene eseguito a runtime (dynamic typing). Una variabile è un contenitore a cui viene associata un'etichetta (il nome) che può essere associata a diversi contenitori anche di tipo diverso durante il suo tempo di vita. Fa parte di Python un sistema garbage collector per liberazione e recupero automatico della memoria di lavoro.

2.2.2 Tensorflow

TensorFlow è una libreria software open source per il machine learning, che fornisce moduli per la definizione e calcolo di operazioni con tensori.
I tensori sono una struttura dati generale definita da un singolo spazio vettoriale, come lo sono i vettori e le matrici.

Questo framework, largamente utilizzato in ambito di ricerca scientifica, sta alla base di molti di prodotti commerciali Google come il riconoscimento vocale, Ricerca, Gmail e Google Foto.

2.2.3 Keras e Scikit-learn

Keras è un'interfaccia per la programmazione di framework di apprendimento automatico che permette di definire e addestrare un modello di deep learning in pochi passaggi. Tra i framework supportati da Keras figura TensorFlow. Keras fa da front-end per la costruzione di modelli e Tensorflow effettua il lavoro di training.
Prima è necessario definire i dati per il training: tensori di input e tensori con i target. La fase successiva è definire una rete di strati che mappano gli input con i loro output. Usando Keras non bisogna preoccuparsi della compatibilità della forma dei tensori tra i vari strati, perché gli strati che vengono aggiunti sono costruiti in modo da avere la stessa forma dello strato successivo. I layer formano delle pipeline di trasformazioni dei dati.

In seguito deve essere configurato il processo di apprendimento scegliendo la funzione di loss, una funzione ottimizzatore e i parametri per monitorare l'addestramento.

Creata la rete, vengono iterati i training data chiamando la funzione *fit*.

Con Keras lo stesso codice può essere eseguito sia su CPU che su GPU.

Per definire un modello ci sono due approcci:

- usare la classe Sequential (solo per strati sequenziali)

- usare il functional API (per grafi aciclici di strati, che permette di costruire architetture del tutto arbitrarie). Prima vengono manipolati i tensori che il modello processerà, poi a questi tensori vengono applicati gli strati come se questi fossero funzioni.

Scikit-learn è una libreria Python open source per l'apprendimento automatico. Supporta algoritmi all'avanguardia come decision tree e random forest. È costruito basandosi su

NumPy. Scikit-learn aiuta nel pre-processing, nella riduzione della dimensionalità (selezione dei parametri), nella classificazione e nella regressione.

2.2.4 NumPy

NumPy è un supporto per array multi-dimensionali.
NumPy è una libreria di Python che nasce per lavorare con gli array. Fornisce anche funzioni per effettuare trasformazioni e lavorare con matrici.

NumPy fornisce operazioni per manipolare gli array come la NumPy slicing operation che permette di selezionare valori corrispondenti agli indici indicati tra parentesi quadre.

Un'altra operazione a disposizione è il reshape dell'array e consiste nel riarrangiare le righe e le colonne.

NumPy significa "Numerical Python".
Si preferisce usare NumPy invece delle liste di Python perché questa libreria fornisce array 50 volte più veloci. Questo è possibile grazie alla memorizzazione continua degli array, al contrario delle liste e possono essere processati e manipolati in modo più efficiente. Questa tecnica è chiamata località di riferimento.

2.2.5 Pandas

Pandas è una libreria di Python che fornisce strutture dati e strumenti per la manipolazione ed estrazione di dati ed è impiegata nel data analysis e machine learning.

Questa libreria è ideale per gestire dati tabulari ed è costruita basandosi su NumPy.

Pandas è utile per trasformazione dei dati, riempimento delle tabelle con dati mancanti, normalizzazione dei dati, unione di tabelle, simulazioni numeriche, modelli statistici, visualizzazione dei dati, e machine learning.

2.3 Modelli di previsione

In questa sezione verranno fornite le informazioni sui modelli di previsione impiegati nell'analisi riportata nel Capitolo 3. In particolare si approfondiranno i concetti di rete neurale, neuroni artificiali, Random Forest, Decision Tree. La bibliografia di riferimento è F. Chollet, "Deep Learning with Python", Shelter Island: Manning Publications Co, 2018.

La sitografia di riferimento per Random Forest e Decision Tree è https://www.ibm.com/ e https://careerfoundry.com/.

2.3.1 Reti neurali

Le reti neurali sono una sequenza di strati (o layer) che elaborano i dati per restituire una previsione.

Il deep learning è un framework matematico per l'apprendimento di rappresentazioni dai dati.

In Figura 2.4 è riportata un'immagine che raffigura come una rete a più strati trasforma l'immagine di un numero scritto a mano per riconoscere qual è il numero raffigurato.

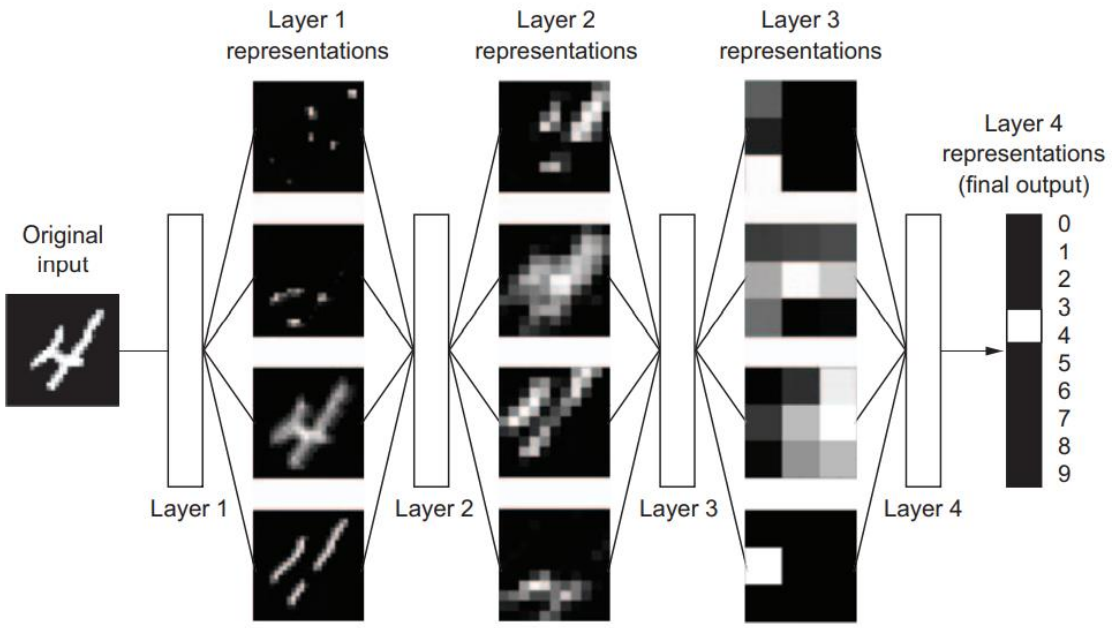

Figura 2.4 Rappresentazione di una rete a più strati che apprende la classificazione delle cifre

La rete trasforma l'immagine in rappresentazioni che si avvicinano sempre di più al risultato finale e si allontanano dall'immagine originale.

La rete neurale è come un filtro che distilla l'informazione fino a ottenere solo le informazioni utili per il task.

La specifica delle operazioni che ogni strato applica ai dati di input è raccolta nei pesi associati a ogni layer, come mostrato nella Figura 2.5. La trasformazione implementata da uno strato è parametrizzata dai suoi pesi. L'apprendimento consiste nel trovare un insieme di pesi che permettono alla rete di mappare correttamente gli input con i tag. Il problema è che una rete può contenere milioni di pesi, quindi risulterebbe difficile trovare i valori corretti, soprattutto perché la modifica di un parametro cambia il comportamento degli altri.

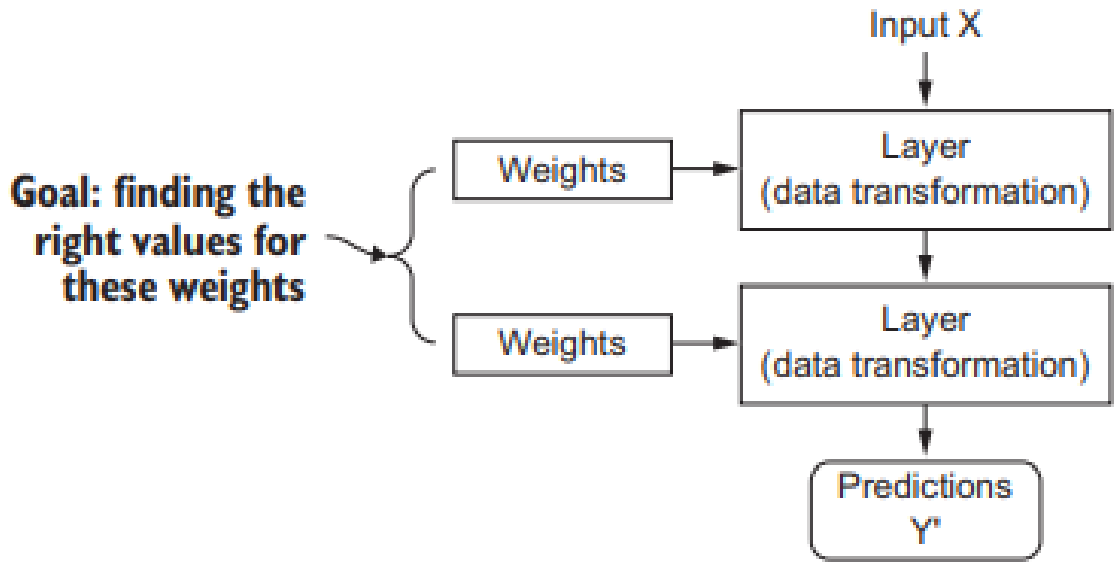

Figura 2.5 Una rete neurale parametrizzata dai suoi pesi.

Per migliorare la rete bisogna sapere quanto è performante, per questo vengono utilizzate le funzioni di perdita (*loss function*). Questa misura la lontananza tra la predizione della rete e il tag corretto.

Il risultato viene impiegato come segnale di feedback per aggiustare il valore dei pesi diminuendo l'errore. Il cambiamento dei pesi viene affidato all'ottimizzatore che implementa l'algoritmo di Backpropagation.

Inizialmente i pesi della rete sono casuali e il primo risultato della funzione di perdita è molto alto, ma la rete continua a processare nuovi dati e aggiustare i pesi diminuendo l'errore. Le reti con un errore minimo hanno l'output molto vicino a quello corretto.

I layer all'interno di una rete si suddividono in input, nascosti e output.
Ogni livello è composto da un certo numero di neuroni, deciso in fase di costruzione della rete.

Le unità nascoste (o neuroni) indicano la dimensione dello spazio di rappresentazione. Avendo più unità nascoste, lo spazio di rappresentazione diventa più grande e permette alla rete di imparare rappresentazioni più complesse, con un costo computazionale maggiore. Se lo spazio di rappresentazione fosse troppo grande, la rete potrebbe imparare pattern non voluti, che migliorano le performance con i training data, ma non con i test data.

Ogni strato può contenere uno o più neuroni. L'output di ogni neurone viene mandato in input a qualche neurone del livello successivo, oppure a tutti se la rete è *densa.*
Il layer di input colleziona i dati, i layer nascosti le elaborano, e lo stato di output restituisce il risultato.

La rete neurale riceve dati di input e i corrispondenti output corretti per la fase di addestramento. Questo momento è fondamentale perché la rete impara ad associare gli input al risultato corretto.

Nella fase di testing si chiede alla rete di produrre predizioni usando dati di input mai visti e si verifica la percentuale di predizioni uguali ai risultati corretti.

La rete neurale è principalmente formata da strati che possono essere considerati come dei filtri che processano i dati. Gli strati estraggono rappresentazioni dei dati sempre più significative.

Costruire la rete consiste nel mettere in sequenza più layer, decidere quanti strati e unità nascoste utilizzare. Questo passaggio è molto semplice grazie all'impiego di Keras.

Se bisogna svolgere una classificazione in N classi, la rete deve avere l'ultimo strato con N neuroni.

Una classificazione è definita binaria se a ogni dato di input deve essere associato una delle due categorie considerate. Se la classificazione fosse multi-classe, le categorie in cui suddividere i dati di input sarebbero più di due.

Dopo aver scelto quanti layer utilizzare e quante unità per ognuno, bisogna scegliere la funzione di perdita, l'ottimizzatore, la metrica.

La loss function viene impiegata per misurare la performance della rete neurale.
La loss function *crossentropy* è adatta nei casi in cui l'output della rete è una probabilità. Crossentropy misura quanto si discostano tra loro le distribuzioni di probabilità corrette e quelle calcolate dalla rete.
Si utilizza *binary_crossentropy* se la classificazione è binaria, categorical_crossentropy se la classificazione è multi-classe.

L’ottimizzatore aggiorna i pesi della rete. I pesi contengono le informazioni raccolte con l’addestramento. Inizialmente i pesi hanno valori casuali poi vengono aggiustati gradualmente basandosi sui feedback e in base all’ottimizzatore. L’ottimizzatore *rmsprop* generalmente è adatto per qualsiasi problema.

La metrica può essere l’accuratezza o un altro parametro che misura la performance della rete.

Ora la rete può essere addestrata con il metodo fit fornito da Keras che esegue vari training loop.
Nel metodo fit deve essere indicato quante epoche eseguire. Le epoche sono iterazioni del training loop sugli stessi dati.

I cicli di training sono formati da vari step.
Prima viene creato un batch di dati con il loro output corretto. I modelli di deep-learning non processano tutti dati in una volta, ma questi vengono divisi in batch.

I batch sono un piccolo insieme di dati (tipicamente tra 8 e 128) che vengono processati simultaneamente dal modello. Il numero di sample (dati forniti alla rete) è di solito una potenza di 2 per facilitare l’allocazione di memoria della GPU. Ogni batch ha la sua discesa del gradiente e i suoi pesi.

La seconda fase è l’esecuzione della rete su un batch di dati e viene restituito un output.

Con il risultato viene misurata la distanza dall’output corretto con la metrica indicata precedentemente.

Come ultimo step vengono aggiornati i pesi per diminuire l'errore.
Un approccio per compiere questa fase è sfruttare il fatto che tutte le operazioni applicate sono differenziabili e calcolare il gradiente dell'errore e poi spostare i coefficienti nella direzione opposta al gradiente per diminuire l’errore.
L’algoritmo Backpropagation, partendo dallo strato in cima scendendo fino allo strato più basso, calcola quanto ogni parametro contribuisce al valore di perdita applicando la regola della catena.

La regola della catena è una regola di derivazione che permette di calcolare la derivata della funzione composta di due funzioni derivabili:

f(g(x)) = f'(g(x)) * g'(x).

Si calcola la derivata della funzione di perdita. La derivata rappresenta l'andamento della funzione ed è positiva se la funzione è crescente. L'obiettivo è quello di diminuire l'errore quindi si segue il senso opposto della derivata.

Se la funzione data è differenziabile, è possibile trovare il minimo della funzione analiticamente. Il minimo di una funzione è il punto in cui la derivata è 0. Bisogna quindi trovare i punti in cui la derivata si azzera e controllare i punti con i valori minimi.

Per esempio W -= step * gradient

Bisogna prestare attenzione a scegliere un valore ragionevole per il fattore step. Se è troppo piccolo, serviranno molte iterazioni per scendere lungo la curva e ci si può fermare a un minimo locale. Se lo step è troppo grande, l'aggiornamento dei pesi porta ad andare in punti casuali della curva.

Si osserverà che l'errore tra output corretto e calcolato diminuisce a ogni iterazione e l'accuratezza aumenta (F. Chollet, "Deep Learning with Python", Shelter Island: Manning Publications Co, 2018, pp. 51-52).

.

Il numero di epoche è importante perché un numero di iterazioni troppo grande porta all'overfitting.
Il fenomeno di overfitting si verifica quando il modello tende a comportarsi peggio su nuovi dati rispetto ai dati di training. La rete viene ottimizzata sui dati di training e impara la rappresentazione di dati specifici perdendo in generalizzazione. La soluzione è diminuire il numero di epoche e tenere monitorata la performance della rete su dati mai visti, cioè quelli per il test. Inoltre, se si hanno a disposizione pochi dati, è preferibile usare una rete di piccole dimensioni con pochi neuroni (1 o 2) per evitare l'overfitting.

La funzione di attivazione è necessaria per ampliare lo spazio di ipotesi.
Relu (rectified linear unit) è la funzione più conosciuta in deep learning ed è adatta a risolvere molti problemi. Questa funzione porta a zero i valori negativi.

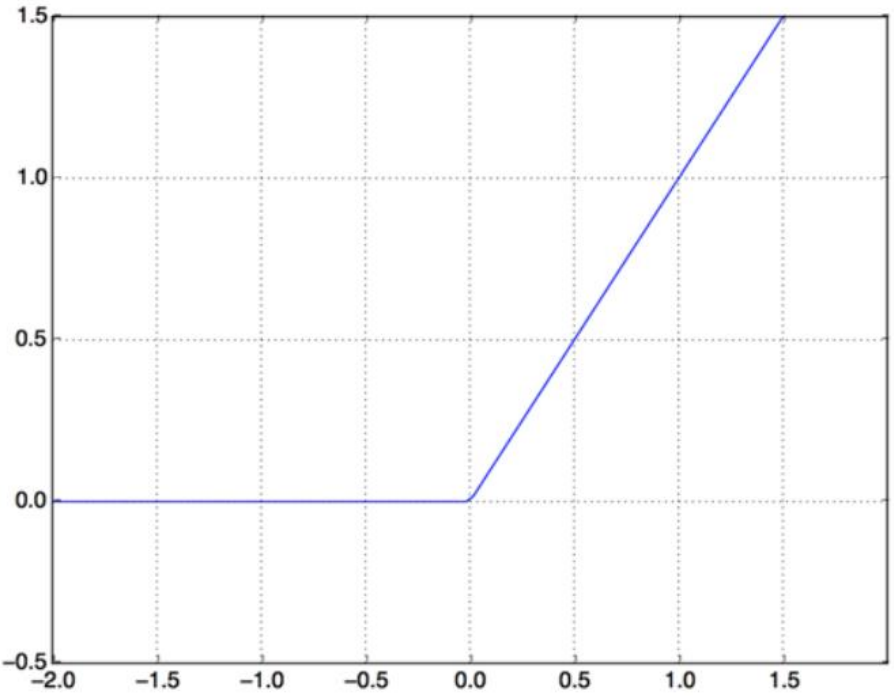

Figura 2.6: Funzione relu

Nel caso di classificazione binaria è consigliato usare nell'ultimo layer la funzione *sigmoide*. Questa funzione restringe i valori in un intervallo [0,1].

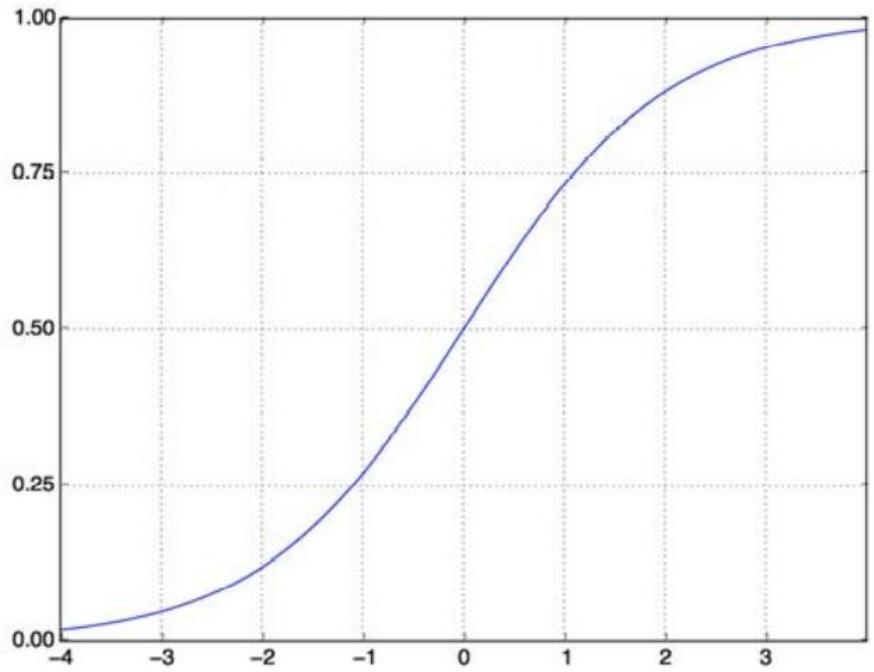

Figura 2.7: Funzione sigmoide

Se la classificazione fosse multi-classe, la rete dovrebbe finire con la funzione softmax.

Fino ad ora si è discusso solo di classificazione, ma un altro problema tipico del machine learning è la regressione.

La classificazione individua l'appartenenza ad una classe. Per esempio un modello potrebbe predire come il potenziale cliente risponderà a un'offerta. Con la classificazione l'output predetto (la classe) è categorico ossia può assumere solo pochi valori.

La regressione predice un valore numerico specifico. Ad esempio un modello potrebbe predire che il cliente porterà un certo profitto nel corso di un determinato periodo di tempo. Le variabili in uscita possono assumere un numero illimitato (o comunque una grande quantità) di valori. Spesso queste variabili in uscita sono indicate come continue anche se talvolta non lo sono nel senso matematico del termine (ad esempio l'età di una persona).

La regressione consiste nel predire valori continui invece di classi rappresentate da valori discreti.

Un esempio di regressione è la predizione delle temperature basata sui dati metereologici oppure il tempo necessario per un progetto software partendo dalle sue specifiche.

Per la regressione la funzione di loss più utilizzata è l'errore quadratico medio (MSE). Anche la metrica per la validazione differisce da quelle usate per la classificazione perché l'accuratezza non è calcolabile con questo tipo di problemi. Solitamente viene scelto l'errore assoluto (MAE).

Per valutare la rete neurale bisogna dividere i dati che abbiamo a disposizione in tre parti: i dati per l'addestramento (training data), per la validazione (validation data), per il test (test data).

É importante la valutazione della rete perchè lo sviluppo di un modello consiste anche nell'adattare le sue configurazioni. Queste vengono modificate con gli iper-parametri che sono ad esempio il numero di strati o la dimensione degli strati. Questi valori sono aggiustati utilizzando come guida il segnale di feedback fornito dalla fase di validazione del modello.

2.3.2 Random forest

Random Forest è un algoritmo di ML che combina l'output di più decision tree (DT) per ottenere un singolo risultato. Questo modello di previsione alternativo è costituito dall'aggregazione di DT.

Alberi di decisione

L'algoritmo di DT è formato da più domande che formano i nodi di decisione che suddividono i dati. La decisione finale è denotata dal nodo foglia. Un esempio di DT è mostrato nella Figura 2.8.

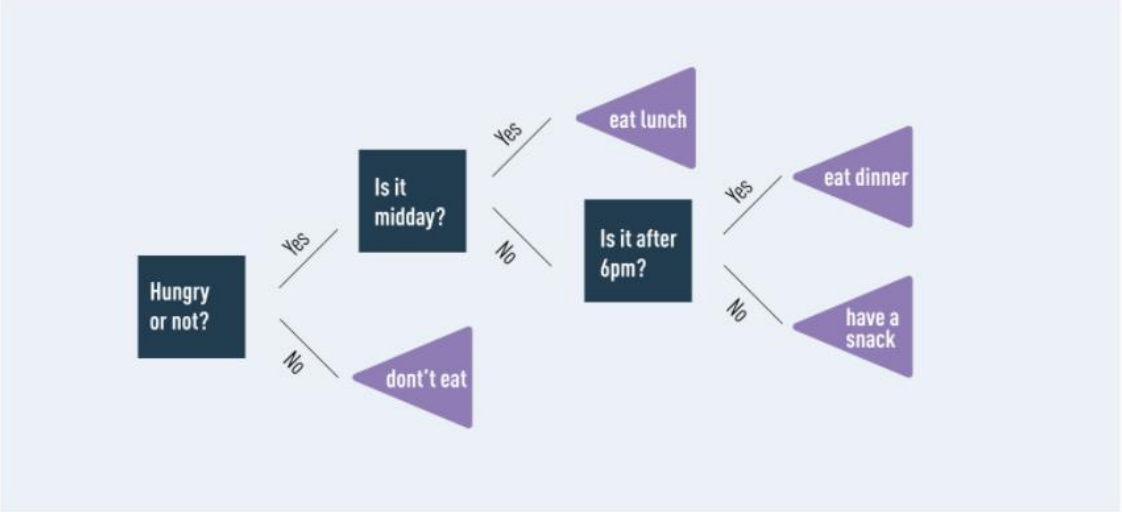

Figura 2.8: esempio di albero di decisione

Quando diversi alberi di decisione vengono raggruppati in un algoritmo di Random Forest (RF) la predizione ha un'accuratezza maggiore.
La logica dietro il modello RF è che diversi alberi di decisione non correlati si comportino meglio in gruppo piuttosto che da soli.
Quando si usa RF per una classificazione, viene scelto il risultato con più voti, cioè quello più volte restituito dai vari alberi di decisione. Nel caso di un problema di regressione viene fatta una media degli output di tutti gli alberi.
Dato in input il dataset con le feature e gli output corretti, il DT formula delle regole che lo aiuteranno a fare predizioni. RF invece seleziona solo un sottoinsieme casuale delle feature, in modo da avere alberi poco correlati tra loro, e poi fa una media dei risultati.

Addestramento

L'algoritmo RF prima di essere addestrato deve avere 3 iper-parametri impostati. Questi includono la dimensione dei nodi, il numero di alberi e il numero di feature dati in input.

Durante l'addestramento i pesi associati a ogni feature vengono aggiustati.
Le feature sono caratteristiche o parametri di input. Nelle reti neurali i pesi sono nascosti, nelle RF è possibile sapere le feature che sono più importanti e determinanti per la creazione dell'output.

Questo algoritmo può essere utilizzato sia per problemi di classificazione sia di regressione come con le reti neurali.
I DT nell'algoritmo RF vengono addestrati con il metodo ensemble *bagging*. Un metodo ensemble è una tecnica che combina le previsioni di più algoritmi di apprendimento automatico per fare previsioni più accurate rispetto a qualsiasi singolo modello. Bagging è l'applicazione del metodo *bootstrap* a una grande varietà di algoritmi di ML.

Bootstrap esegue casualmente il campionamento delle righe e delle feature dal dataset di partenza per formare un dataset di esempio per ogni modello. Questo metodo riduce la varianza di algoritmi come gli alberi di decisione.

La varianza è l'errore conseguente alla sensibilità a piccole perturbazioni nel dataset usato per l'addestramento. Un'alta varianza porta l'algoritmo a modellizzare dati irrilevanti. Questo problema è l'overfitting che causa una migliore performance dell'algoritmo sui dati di training, non distinguendo tra dati rilevanti e irrilevanti.

L'utilizzo delle RF ha diversi vantaggi:
1. Riduce il rischio di overfitting.
2. Flessibilità: RF può gestire sia problemi di regressione che classificazione, mantenendo un'accuratezza alta.
3. Facilità con cui si può valutare l'importanza o contributo di una variabile al modello. Il criterio *gini* misura quanto l'accuratezza del modello diminuisce quando una variabile viene esclusa.

3. Analisi dei dati

In questo capitolo è spiegata la preparazione dei dati, l'elaborazione e i risultati ottenuti impiegando modelli di machine learning presentati nel capitolo precedente.

Utilizzando i modelli predittivi e i dati a disposizione è stata calcolata la condizione del paziente dopo 5 anni, variando i dati impiegati. Sono stati inoltre individuate le feature più determinanti nel calcolo dell'output e confrontate le accuratezze dei modelli creati con l'accuratezza degli stadi e del parametro T del TNM.

3.1 Preprocessing

I dati forniti dall'Azienda Ospedaliera di Parma, prima di poter essere utilizzati dai modelli di previsione, devono essere sottoposti a preprocessing.

Il preprocessing dei dati ha l'obiettivo di strutturare i dati per essere forniti in input alle reti neurali. Questa pratica comprende la vettorizzazione, la normalizzazione e la gestione di valori mancanti.

Vettorizzazione
Tutti gli input delle reti neurali devono essere tensori di numeri interi o in virgola mobile.

I tensori sono array multi-dimensionali forniti dalla libreria NumPy e vengono impiegati come struttura dati di base nel machine learning. La dimensione dei tensori è chiamata *asse.*
Un tensore che contiene un solo valore è definito *scalare* e ha zero assi. Il numero di assi in un tensore è anche chiamato *rank.*

Un tensore è definito da tre attributi:

- il rank è il numero di assi e in NumPy l'attributo si chiama *ndim.*

- la forma è la dimensione lungo ogni asse, esplicitata da una tupla di interi. In NumPy è contenuta nell'attributo *shape.*

- il tipo è il tipo dei dati contenuti nel tensore. In Python l'attributo *dtype* può essere float32, uint8, float64.

Audio, immagini o testi, per essere processati, devono prima essere tradotti in tensori attraverso la vettorizzazione. Nel caso di testi si può creare una matrice di codifica. Ogni indice di colonna corrisponde a una parola specifica e su ogni riga viene codificata una parola settando a 1 la cella con l'indice di colonna corrispondente. Questa tecnica di vettorizzazione è chiamata one-hot encoding e permette di creare tensori di 0 e 1. Le immagini possono essere tradotte in matrici 4D di interi. I video vengono raccolti in tensori 5D.

Oltre ai dati di input devono essere vettorizzati anche gli output utilizzati per l'addestramento.

Normalizzazione

La normalizzazione consiste nel ridurre i dati di input a un range per facilitare l'apprendimento della rete. Bisogna evitare di fornire alla rete valori molto grandi e dati eterogenei.

In generale è meglio utilizzare valori piccoli nell'intervallo [0,1] e dati omogenei cioè compresi nello stesso range anche se dati di feature diverse.

Gestione dati mancanti

Con le reti neurali si possono sostituire i dati mancanti con '0', se questo non è un valore significativo.

La rete neurale capirà che 0 significa che il valore è mancante e inizierà a ignorarli.

Con i dati forniti dall'ospedale di Parma è stato necessario tradurre alcune parole in numeri, ma soprattutto cancellare righe di pazienti che avevano pochissimi dati oppure cancellare colonne vuote senza nessun valore. Per contenere in una struttura dati le

tabelle Excel con i dati dei pazienti e manipolare i dati è stato utilizzato Pandas. Utilizzando sempre i dataframe di Pandas, le tabelle con tipologie di dati diverse sono state unite grazie alla funzione merge.

Una volta che i dati erano pronti, sono stati convertiti in tensori NumPy.
Nella classificazione dei pazienti nei due stati 'vivo' e 'morto', non c'è stato bisogno della vettorizzazione, mentre nella classificazione in tre stati l'output è stato sottoposto alla tecnica di one-hot encoding utilizzando la funzione fornita da Keras *to_categorical*.

3.2 Realizzazione Reti Neurali e Random Forest

Partendo dai dati forniti dall'AOP, sono stati realizzati vari modelli di reti neurali e random forest che calcolano previsioni sulla prognosi del paziente. Questi modelli in input prendono dati di tipo diverso, ma l'output che si vuole calcolare è sempre il decorso della malattia a 5 anni dalla prima diagnosi.
L'output è codificato con i numeri 0 e 1 che rispettivamente rappresentano il decesso e la vita del paziente.
La previsione corretta è calcolata partendo dalla tabella dei dati clinici che contiene la data della prima diagnosi e la data del decesso, se avvenuto. Il calcolo dell'output corretto è contenuto in una funzione denominata *compute_alive_after,* che prende in input il dataframe di Pandas, estrae la colonna con la data della prima diagnosi e la colonna con la data del decesso.
Se la data di morte non è presente, significa che il paziente è ancora vivo quindi in corrispondenza del suo ID verrà settato il valore 1 nel vettore NumPy, variabile di ritorno della funzione. Se il paziente è deceduto e la morte è avvenuta entro 5 anni, questo caso è codificato con 0, altrimenti 1.
La funzione restituisce un vettore NumPy di 0 e 1 che rappresenta l'output corretto. Una parte di questo vettore verrà utilizzata per il training e una parte per il test dell'accuratezza.

La prima rete neurale che è stata creata prende in input i dati clinici.
La tabella contenente i dati clinici ha molti valori scritti in linguaggio naturale, di conseguenza è necessario il preprocessing. Ove possibile i valori sono stati codificati

con numeri, le righe o colonne con pochi valori sono state eliminate utilizzando Pandas. I pazienti raccolti nei dati clinici sono 92 e le colonne sopravvissute al preprocessing sono 62.

Con il supporto di Keras è stata costruita la rete neurale densa e sequenziale. Sequenziale è una rete che ha gli strati sono messi in sequenza come uno stack (Figura 3.1).

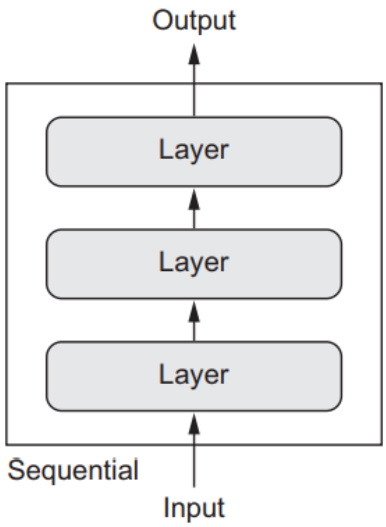

Figura 3.1: Un modello sequenziale: uno stack di strati

Con Scikit-learn sono stati suddivisi i dati a disposizione tra training data e test data. Generalmente l'80% dei dati è stato utilizzato per l'addestramento e il restante per i test di accuratezza.
La rete neurale che utilizza i dati clinici è composta da 3 strati. I primi due hanno 16 unità e la funzione di attivazione *relu*, mentre l'ultimo strato ha solo un'unità nascosta, perché l'output è un solo valore e la funzione utilizzata è *sigmoid* che permette di ridurre il risultato a un valore tra 0 e 1.
L'ottimizzatore scelto è *rmsprop* e la funzione di loss è *binary_crossentropy*, perché la classificazione è binaria.
Il numero di epoche e la dimensione dei batch è stata decisa dopo alcune prove per tentare di minimizzare l'overfitting.

L'accuratezza raggiunta è del 50% probabilmente per la scarsità dei dati a disposizione. Infatti solo 62 pazienti sono rimasti dopo il preprocessing e sono stati considerati in

questa analisi.
L'accuratezza calcolata è la media tra le accuratezze di 100 reti neurali costruite ed addestrate allo stesso modo.

Per creare i modelli di classificazione basati sull'algoritmo random forest, è stata impiegata la libreria Scikit-learn e *gini* come criterio per calcolare il peso delle feature. L'accuratezza ottenuta con i dati clinici è del 63%.

In seguito i modelli predittivi sono stati applicati ai dati clinici raccolti dal progetto europeo BD2Decide, non solo a quelli dell'AOP.

Come mostra la Figura 3.2, utilizzando tutti questi dati clinici, che comprendono 1086 pazienti, l'accuratezza della rete neurale sale al 55% e quella di random forest al 70%.

Aumentando i dati, l'accuratezza sia delle Random Forest sia delle reti neurali aumenta. Un problema generale in questa analisi è la scarsità dei dati, ma se sarà possibile averne a disposizione di più, l'accuratezza aumenterebbe come si può vedere da questo esempio.

Questi risultati ci portano all'ipotesi che i modelli applicati potrebbero condurre a più interessanti e rilevanti risultati predittivi.

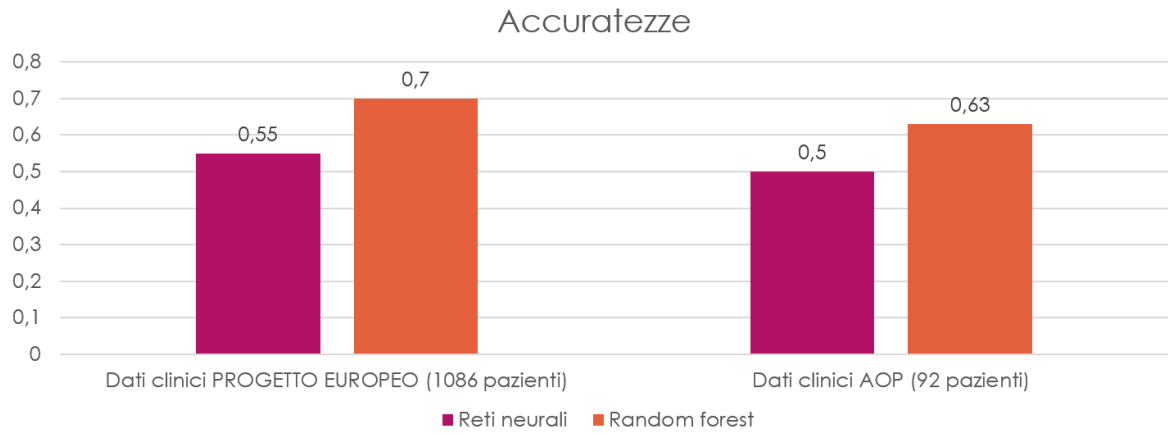

Figura 3.2: Confronto accuratezze tra modelli con dati clinici del progetto europeo BD2Decide e con dati dell'AOP

I dati genomici sono suddivisi in due tabelle diverse: una per i tumori orofaringei e l'altra per i tumori alla cavità orale.
Le tabelle hanno solo valori numerici a parte l'ID del paziente, quindi è solo stato

necessario ordinare i pazienti nello stesso ordine dei dati clinici in modo da avere l'output corretto corrispondente.
I valori non presenti, cioè NaN, sono stati sostituiti con 0.
NaN (Not a Number) indica un insieme di possibili valori non rappresentabili in virgola mobile, cioè valori indeterminati, oppure risultati di operazioni che producono un errore.
Il dataframe Pandas ottenuto viene poi convertito a matrice Numpy per poi essere utilizzato dalla rete.
La rete neurale costruita ha solo 2 strati con pochi neuroni perché i dati sono pochi ed è meglio ridurre la dimensione della rete.
La Figura 3.3 mostra che i pazienti della tabella della cavità orale sono 136, quello dell'orofaringe sono 11 ed entrambe hanno 26904 feature (Figura 3.4).

La Figura 3.5 riporta le accuratezze dei vari modelli con diversi dataset di input.
L'accuratezza con i dati della cavità orale è del 58% con la rete neurale, 62% con random forest.
L'accuratezza con i dati dell'orofaringe è del 50% con la rete neurale, 54% con random forest, ma, essendo i pazienti con tumore all'orofaringe solo 11, questa analisi non è molto attendibile.
Dato che i pazienti orofaringei raccolti sono solo 11, le tabelle sono state unite con la funzione merge di Pandas.
L'accuratezza risultante della rete neurale è 52%, della random forest 61%.

I dati radiomici sono suddivisi in immagini MRI e CT.
Dalle tabelle sono state eliminate le righe che hanno stringhe come valori con la funzione drop di Pandas e le righe con almeno 50 NaN con la funzione dropna.
Nella tabella MRI sono raccolti 224 pazienti, nella tabella CT 69 e in entrambe ci sono 1024 colonne.
L'accuratezza con immagini MRI è del 50% con le reti neurali (NN), 49% con RF, quella con immagini CT 53% con RN e 61% con RF.
Unendo le tabelle l'accuratezza è del 52% sia con NN sia con RF.

Come ultimo modello in input viene fornita l'unione dei dati genomici e radiomici in un unico dataframe Pandas.

L'accuratezza ottenuta è del 51% impiegando le NN e 61% con le RF.

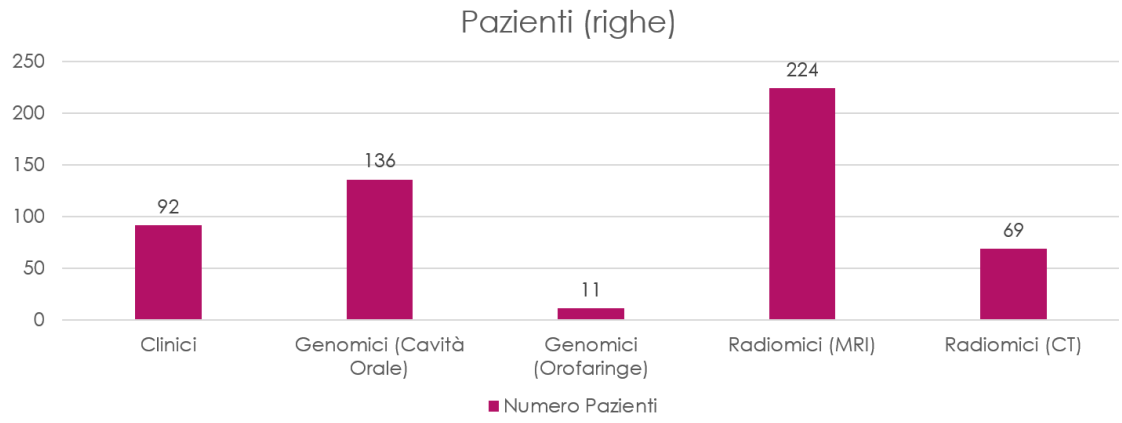

Figura 3.3: Numero di pazienti nei vari dataset

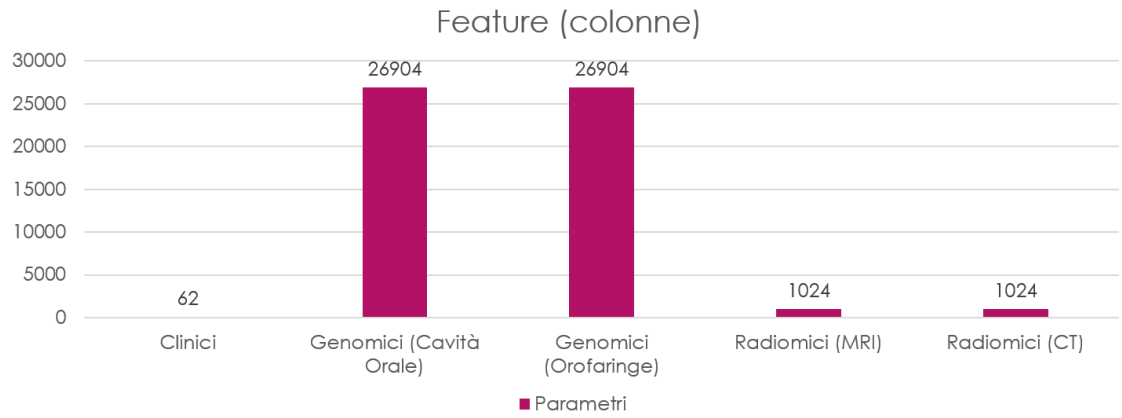

Figura 3.4: Numero di feature nei diversi dataset

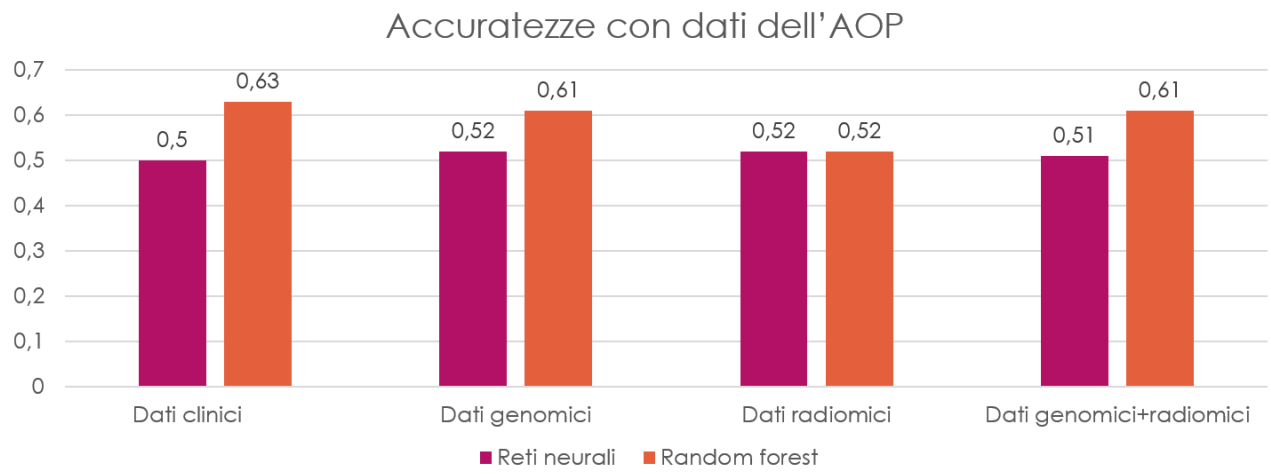

Figura 3.5: Accuratezze delle reti neurali e random forest con i diversi tipi di dato

Per valutare quanto sono significativi i dati clinici, questi sono stati uniti ai dati genomici, entrambi dei pazienti dell'AOP. I risultati sono mostrati nella Figura 3.6. L'accuratezza di NN è 50% e di RF è del 67% con i geni dell'orofaringe, mentre con quelli della cavità orale la rete neurale è accurata nel 48% dei casi, mentre random forest nel 60% dei casi.

Unendo i dati clinici, genomici e radiomici si ottiene un'accuratezza del 55% nelle NN e del 60% nelle RF.

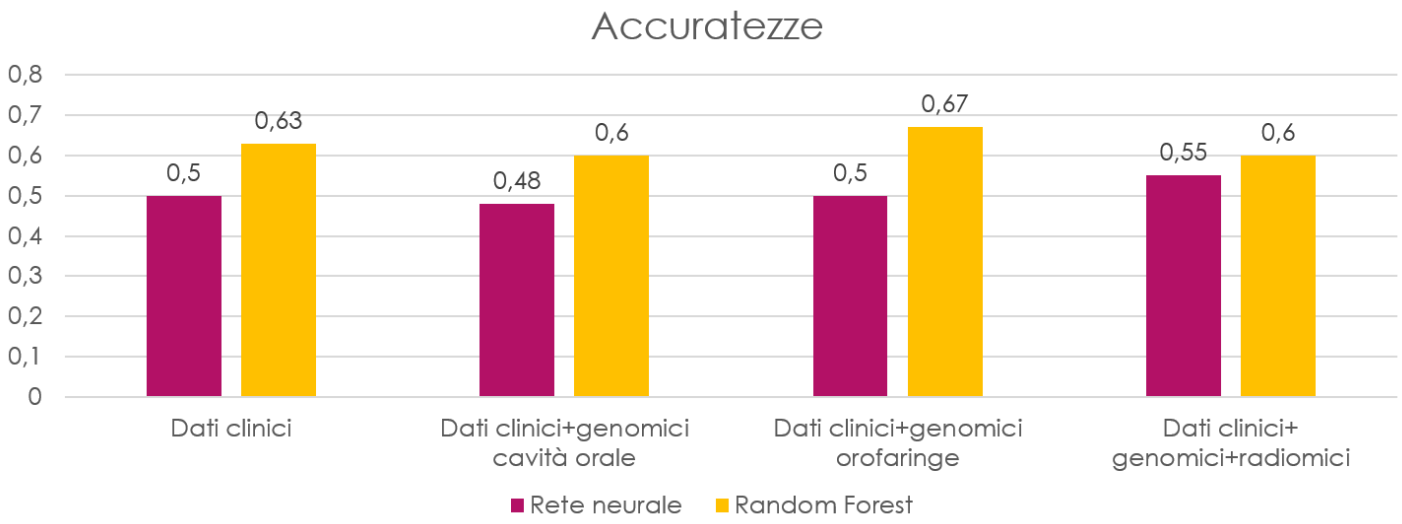

Figura 3.6: Confronto accuratezze con diversi dati

Dalla Figura 3.6 si può notare che i dati radiomici non sono molto significativi e i più determinanti sono i clinici.

3.3 Feature più importanti

Sfruttando il vantaggio delle RF di poter calcolare l'importanza delle diverse feature, sono state calcolate le variabili più significative su 100 modelli distinti, ma creati allo stesso modo.

In una tabella Pandas, per ogni modello creato, è stata inserita una colonna con tutti i pesi associati a ogni variabile identificata dalla riga. Una volta completata la tabella con 100 colonne, la funzione mean ha calcolato la media dei pesi calcolata su ogni riga in modo da avere il peso medio associato a ogni feature.
Questi passaggi sono stati completati per dati clinici, genomici e radiomici.
Per ridurre la dimensione del risultato sono state considerate solo le 50 variabili più importanti.
I risultati ottenuti sono i seguenti:

I dati clinici più importanti sono:

Feature	**Peso in percentuale**
clinical_Age_at_Diagnosis	4,5754
clinical_Year_of_Birth	4,3778
qol_Deglutition	3,9973
clinical_HB_Level	3,6587
clinical_PLT_Level	3,5687
clinical_PLT_Level_Total	3,5669
ctn_TNM_cN_8Edition_OralCavity_Oropharynx_p16Negative_Hypopharynx_Larynx	3,4172
clinical_HB_Level_Total	3,2441
clinical_Leukocytes	3,1059
risk_Pack_Years	3,1009
clinical_Leukocytes_Total	3,0902

risk_Years_as_a_Smoker	2,8459
ctn_Anatomical_Tumor_Location	2,7911
ctn_TNM_cT_8Edition_OralCavity_Oropharynx_p16Negative_Hypopharynx_Larynx	2,7156
ctn_Stage_at_Diagnosis_8Edition_OralCavity_Oropharynx_p16Negative_Hypopharynx_Larynx	2,2074
risk_Packs_Smoked_per_Day	2,1353
risk_Whiskey	2,0028
ctn_Node_Levels_Left	1,9274
ctn_TNM_cN_7Edition	1,9197
clinical_Overall_Comorbidity_Score	1,9098
risk_History_of_Alcohol_Dependence	1,904
ctn_Node_Levels_Right	1,7915
ctn_TNM_cT_7Edition	1,6393
clinical_Lymphocytes_Total	1,5479
clinical_Lymphocytes	1,5394
patho_Degree_of_Cell_Keratinisation	1,539
risk_Oral_Hygiene	1,5203
patho_Grade_at_Diagnosis	1,3832
clinical_ECOG_Performance_Status	1,3487
ctn_TNM_cN_8Edition_Oropharynx_p16Positive	1,3265
INT_OSCC_48gene_model_risk	1,3216
clinical_ASA	1,319
risk_Alcohol_at_Time_of_Diagnosis	1,2155
ctn_Stage_at_Diagnosis_8Edition_Oropharynx_p16Positive	1,2121
risk_Family_History_of_Malignancies	1,0914
ctn_TNM_cT_8Edition_Oropharynx_p16Positive	1,0794
ctn_Tissue_Invasion_ExtraNodalExtension	1,0573
ctn_Tumor_Region	0,9678
ctn_Laterality_of_T	0,9628
INT_HNC_prognostic_LA_model_risk	0,9466

risk_Smoker_at_Time_of_Diagnosis	0,9116
qol_Respiration	0,8954
ctn_Stage_at_Diagnosis_7Edition	0,85
clinical_Hospital	0,842
patho_Basaloid_features	0,7902
risk_Std_Wine	0,7676
ctn_Extension_to_lingual_surface_of_epiglottis_is_present	0,7476
clinical_Sex	0,6752
INT_Lohavanichbutr_OSCC_HPVneg_risk	0,6508
clinical_Zubrod_Performance_Status	0,6402

Tra i dati più importanti figurano l'età del paziente, il TNM, se il paziente è fumatore, se beve alcol, la familiarità, lo stadio. Nella Figura 3.7 è riportato il grafico con le feature cliniche più importanti.

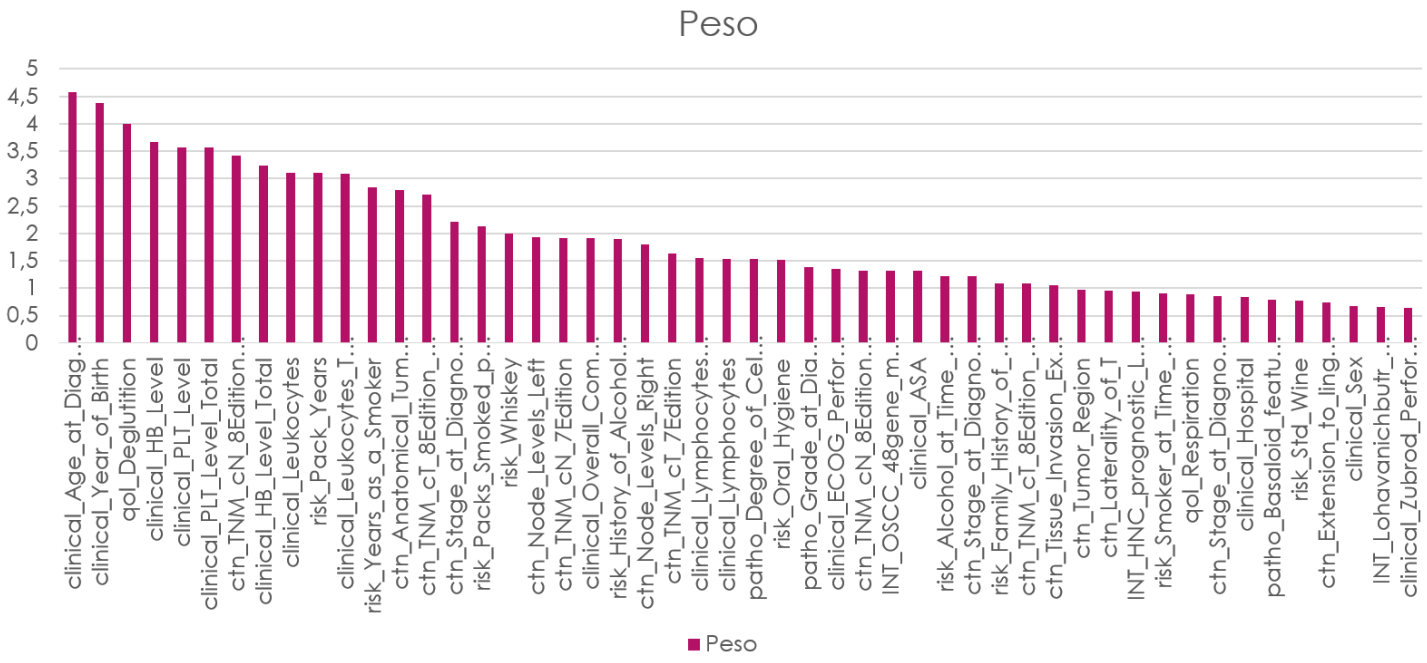

Figura 3.7: Feature dei dati clinici in ordine di importanza

I dati genomici della cavità orale più importanti sono:

Gene	**Peso in percentuale**
SEMA3A	0,1061
C9orf64	0,0915
ZNF785	0,0755
MIR5697	0,0725
LINC01176	0,0712
LOC105372069	0,0652
RAB11B	0,0647
TMEM206	0,0643
CAPG	0,0641
LINC00977	0,061
MIR7854	0,0606
IGHVIII-26-1	0,0575
GUCY2GP	0,0556
GPR26	0,0549
IGHV1-45	0,0543
RPS10P7	0,0534
EIF2S3	0,0533
MZF1	0,0532
KCTD12	0,0526
MTRNR2L2	0,0525
IMMP2L	0,0524
PAXBP1-AS1	0,0503
RHEBL1	0,049
LINC00412	0,049
KRTAP4-11	0,0486
TCHP	0,0474
DTD2	0,0468

GPR32	0,0466
SV2A	0,0463
ZGRF1	0,0435
ALDH7A1	0,0433
IGKV2D-28	0,043
PHPT1	0,0429
FCRL6	0,0418
TCEAL8	0,0418
GPR1	0,041
STOML1	0,041
SLFN11	0,041
LOC100128568	0,0407
ADAM8	0,0405
CDH12P4	0,0403
SOWAHB	0,0403
RAB29	0,0403
UBA3	0,04
KCNK10	0,0399
FANCG	0,0397
IGHVIII-5-2	0,0387
SYNC	0,038
TRIM38	0,0379
HIST3H2BA	0,0374

I dati genomici dell'orofaringe più importanti sono:

Gene	**Peso in percentuale**
RRP12	0,2228
LOC283788	0,1772
RNF165	0,1731
LINC01122	0,1698

LOC100130849	0,1683
OR10G9	0,1581
BANK1	0,153
PHLDA3	0,1448
MIR4525	0,1398
LINC00324	0,1379
RPL23AP78	0,1373
COX7B2	0,1347
PDPR	0,1319
COPS7B	0,1315
RELB	0,1315
LIPE	0,1313
KIAA1586	0,1305
MYH4	0,1304
LOC102467214	0,1291
CNTNAP4	0,1287
SLC25A41	0,1285
SDR42E2	0,1277
APOA1	0,1276
POC1A	0,1257
ACSS3	0,1255
TEF	0,125
SBNO2	0,1231
TMEM117	0,12
TMEM135	0,1198
OR7E126P	0,1191
SEPW1	0,1191
MIR1288	0,1178
MIR4670	0,1171
MT1H	0,1159
MIR4539	0,1159

DBIP1	0,1136
LINC00895	0,1135
TMEM198B	0,111
CROCCP2	0,11
C1QTNF3	0,1099
LDHC	0,1097
MTHFSD	0,1089
MIR6735	0,1086
NARFL	0,108
DBNDD1	0,1079
PRPF4B	0,1064
FDXR	0,1062
NBPF22P	0,1046
DOPEY2	0,1046
MON1B	0,1041

3.4 Risultati dell'analisi genomica

Grazie all'algoritmo Random Forest, è stata calcolata una selezione dei 50 dati genomici che meglio rappresentano e caratterizzano i tumori dei pazienti dell'AOP.

Con questa selezione si è provato a meglio definire la funzione, la locazione, l'espressione e la mutazione più comune di questi geni.

Per identificare i geni è stato utilizzato il database OMIM..

OMIM (Online Mendelian Inheritance in Man) è un compendio di geni umani e fenotipi genetici che è disponibile per tutti e aggiornato giornalmente. Il database contiene più di 16,000 geni.

OMIM è stato creato all'inizio degli anni 60 da Dr. Victor A. McKusick come catalogo dei tratti e disturbi mendeliani, chiamato Mendelian Inheritance in Man (MIM). La versione online, OMIM, è stata creata nel 1985.

Ogni voce di OMIM è data da un identificatore univoco di 6 cifre. La tabella seguente riassume le informazioni ottenute attraverso il database OMIM.

GENE	OMIM	MAPPING	FUNCTION	EXPRESSION
Semaphorin 3a - SEMA3A	* 603961	7q21.11	• neurosignaling; • osteoprotective effect in bones; • signaling in taste receptor cells; • Kallman Syndrome gene (Hypogonadotropic Hypogonadism 16 with or without Anosmia)	Brain Taste receptor cells Bone Thymic cells
Chromosome 9 open reading frame 64 - C9ORF64	* 611342	9q21.32	• Isolated in chromosome 9q deleted in acute myeloid leukemia	-
ZNF785	-	-	-	-
MIR5697	-	-	-	-
LINC01176	-	-	-	-
LOC105372069	-	-	-	-
Ras-associated protein - RAB11B	* 604198	19p13.2	• Part of Ras superfamily of small GTP-binding proteins with regulating exocytotic and endocytotic pathways activities; • Allelic variants were isolated in some neurodevelopmental disorders	-
Proton-activated chloride channel 1 - PACC1 **Or**	* 618427	1q32.3	• Membrane protein functioning as proton-activated chloride channel; • Gene knockout partially protected HEK293 cells from acid-induced death	Brain

Transmembrane protein 206 - TMEM206				
Capping protein, gelsolin-like - CAPG	* 153615	2p11.2	• Modulator of actin structures in response to external stimuli	Ubiquitary
LINC00977	-	-	-	-
MIR7854	-	-	-	-
IGHVIII-26-1	-	-	-	-
GUCY2GP	-	-	-	-
G protein-coupled receptor 26 - GPR26	* 604847	10q26.13	• Sequence homology with serotonin receptor 5-HT5A and gastrin-releasing hormone receptor BB2	-
Immunoglobulin heavy chain variable gene cluster – IGHV1-45	* 147070	14q32.33	• N-terminal variable (V) region containing the antigen-binding site, of immunoglobulin heavy chains; • Genetic variation of these loci may be associated with genetic differences in immune response	-
RPS10P7	-	-	-	-
Eukaryotic translation initiation factor 2, subunit 3 - EIF2S3	* 300161	Xp22.11	• Translation initiation factor; • Mutations causing syndromic X-linked mental retardation (MEHMO S.)	-
Myeloid zinc finger 1 - MZF1	* 194550	19q13.43	• Metal-binding proteins active as transcriptional regulator in myeloid differentiation	-
Potassium channel tetramerization domain-	* 610521	13q22.3	• Potassium channel: N-terminal voltage-gated potassium channel	Strongly expressed in fetal tissues; barely

containing protein 12 - KCTD12			tetramerization (T1) domain	detectable in adult tissues
MTRNR2L2	-	-	-	-
Inner mitochondrial membrane peptidase, subunit 2 - IMMP2L	* 605977	7q31.1	• Mitochondrial membrane protein: inner membrane peptidase	Ubiquitary
PAXBP1-AS1	-	-	-	-
Rheb-like protein 1 - RHEBL1	* 618956	12q13.12	• RAS family small GTPase active in control of growth, differentiation, and transformation • Involved in signaling of mTOR pathway	Ubiquitary
LINC00412	-	-	-	-
KRTAP4-11	-	-	-	-
Trichoplein – TCHP Keratin filament-binding Mitostatin	* 612654	12q24.11	• negative regulator of ciliogenesis, • recruitment of microtubules to centriole controller, • maintenance of mitochondrial morphology • Overexpression inhibites cell growth and enhances apoptotic response • Gene knockdown increased cell survival rate	Liver Kidney Heart Skeletal Muscle Enterocytes Reduced in bladder and breast tumors (advanced stages)
DTD2	-	-	-	-
G protein-coupled receptor 32 - GPR32	* 603195	19q13.33	• Chemoattractant receptors	-
Synaptic vesicle glycoprotein 2A - SV2A	* 185860	1q21.2	• Uptake control of neurotransmitters into vesicles	Highest levels in brain, lung and ovary. Intermediate

			• Insulin secretion regulation • Binding site of botulinum neurotoxin A	expression in heart, kidney, pancreas, spleen and testis
ZGRF1	-	-	-	-
Aldehyde dehydrogenase 7 family, member a1 - ALDH7A1	* 107323	5q23.2	• Semialdehyde dehydrogenase in the acid pathway of lysine catabolism; • Gene mutations in patients with pyridoxine-dependent epilepsy	-

L'obiettivo di queste analisi era individuare le caratteristiche dei pazienti per meglio indentificare un profilo genomico e molecolare come guida alla scelta del trattamento più adatto a ogni paziente.

Le considerazioni riguardo la selezione di geni prodotta sono le seguenti.

Solo 16 dei primi 32 geni analizzati sono nel database OMIM, per gli altri non abbiamo informazioni riguardo l'espressione, funzione, caratterizzazione.

Riguardo i geni di cui abbiamo informazioni, la loro espressione è distribuita nei tessuti umani adulti. Tre geni (PACC1, SEMA3A, SV2A) sono maggiormente espressi nel tessuto cerebrale. Il gene KCTD12 è fortemente espresso nei tessuti fetali ma poco nei tessuti adulti.

Studi di espressione di TCHP (Trichoplein – Keratin filament-binding Mitostatin) (Vecchione et al. 2009) hanno mostrato che la sua sovraespressione in diverse linee cellulari inibisce la crescita cellulare e la formazione di colonie e migliora la risposta della morte programmata delle cellule (apoptosi) a vari stimoli apoptotici . Al contrario, l'abbattimento della mitostatina ha comportato un aumento della crescita cellulare e dei tassi di sopravvivenza cellulare. L'espressione della mitostatina era significativamente ridotta in alcuni tumori primari della vescica e del seno e la sua riduzione era associata a stadi avanzati del tumore; lo studio citato ha concluso che la mitostatina potrebbe essere considerata un soppressore del tumore.

Considerando le funzioni dei geni, abbiamo trovato alcuni punti salienti interessanti.

Ad esempio RHEBL1 è un membro della famiglia RAS di piccole GTPasi attive nel controllo della crescita, differenziazione e trasformazione. Utilizzando l'analisi della sovraespressione nelle cellule HEK293, Tee et al. (2005) hanno mostrato che RHEBL1 umano funzionava a monte dei percorsi MTOR e promuoveva la segnalazione MTOR.

Il gene RAB11B codifica per un membro della sottofamiglia Rab11 GTPasi che si associa specificamente al riciclaggio degli endosomi ed è implicato nella regolazione del traffico vescicolare; fa parte della superfamiglia Ras di piccole proteine leganti GTP (come RHEBL1).

Due geni hanno una funzione diretta nel citoscheletro delle cellule: CAPG, un modulatore delle strutture di actina in risposta a stimoli esterni, e TCHP che è attivo nella regolazione negativa della ciliogenesi, nel controllo del reclutamento dei microtubuli nel centriolo e nel mantenimento della morfologia mitocondriale.

Inoltre, si possono citare due recettori per le proteine G (GPR26 e GPR32), un enzima di membrana mitocondriale con funzione peptidasi (IMMP2L) e due canali ionici di membrana (PACC1 - canale del cloruro attivato dal protone e KCTD12 - canale del potassio voltaggio-dipendente N-terminale dominio).

Potrebbe essere interessante notare che alcuni dei geni analizzati presentano varianti alleliche con chiare e dimostrate relazioni gene-fenotipo. Il primo ad essere notato è SEMA3A (il primo gene della lista in base al peso associato in percentuale) che è responsabile della Sindrome di Kallman - Ipogonadismo Ipogonadotropico con o senza Anosmia.

Poi abbiamo il citato RAB11B correlato al disturbo dello sviluppo neurologico con andatura atassica, discorso assente e diminuzione della sostanza bianca corticale.

La sindrome MEHMO (ritardo mentale, crisi epilettiche, ipogonadismo e ipogenitalismo, microcefalia e obesità) è correlata a mutazioni emizigoti nel gene EIF2S3, che è la subunità centrale del fattore 2 di inizio della traduzione eucariotica (eIF2), coinvolto nel reclutamento di metionil- tRNA(i) alla subunità ribosomiale 40S

essenziale per la sintesi proteica; mentre le mutazioni di ALDH7A1 (enzima della via del catabolismo della lisina) si riferiscono all'epilessia piridossina-dipendente.

3.5 Stratificazione dei pazienti in quattro categorie

La funzione denominata *compute_follow_up* suddivide in 4 categorie i pazienti del dataframe di input in base ai dati del follow up.
Le classi sono 4:

1. AND (Alive without/No Disease) sono i pazienti vivi senza recidive di malattia.

2. AWD (Alive With Disease) sono i pazienti vivi che hanno avuto recidive entro cinque anni.

3. DOD (Dead of Disease) sono i pazienti deceduti per la malattia.

4. DOC (Dead of Other Cause) sono i pazienti deceduti per altre cause.

Per la suddivisione, oltre a controllare la data del decesso nei dati clinici, è stata analizzata la colonna *follow_Recurrence* che riporta 'Yes' in caso di recidiva.
In caso di data del decesso, se nella colonna *follow_Primary_Cause_of_Death* figura il valore 'Other cause', il paziente rientrerà nella categoria DOC.

Ora tutti i pazienti possono essere suddivisi in 3 stati. I pazienti DOC sono stati eliminati dalle analisi perché non sono casi significativi e utili.

La Figura 3.8 riporta la distribuzione di 152 pazienti nelle tre categorie.

Pochi pazienti fanno parte dello stato intermedio.

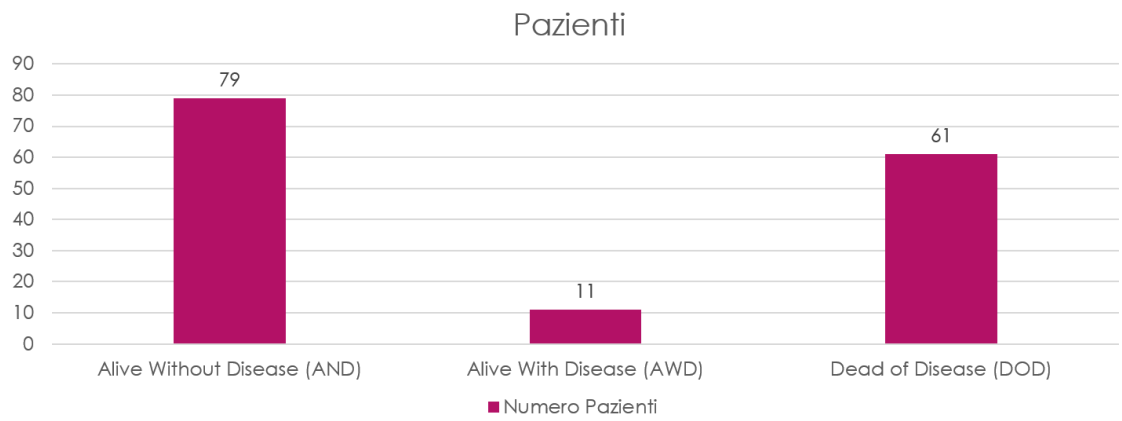

Figura 3.8: suddivisione dei pazienti dell'AOP nelle tre categorie

I modelli di predizione creati in precedenza vengono ricostruiti usando 3 classi di output.

Rispetto agli altri modelli bisogna inserire 3 unità nascoste nell'ultimo strato perché gli output saranno 3. A ogni output corrisponde una categoria e il risultato sarà un vettore di tutti 0 tranne la colonna corrispondente alla classe di cui fa parte il paziente. La funzione di attivazione dell'ultimo strato è *softmax* e la funzione di loss è *categorical_crossentropy*, perché la classificazione è multi-classe.

Il vettore NumPy Y con l'output corretto costituito da 0,1,2 è stato trasformato con la funzione *to_categorical*, fornita da Keras, che sfrutta la tecnica di one-hot encoding e Y è diventato un vettore di 0 e 1.

Al valore 1 corrisponde la categoria di cui fa parte il paziente di quella riga.

L'accuratezza media con i dati clinici è 43% per le NN, 49% per le RF.

L'accuratezza media con i dati radiomici è 50% per le NN, 34% per le RF.

L'accuratezza media con i dati genomici è 50% per le NN, 42% per le RF.

L'accuratezza media con i dati genomici e radiomici è 48% per le NN, 45% per le RF.

Come mostra la Figura 3.9 l'accuratezza dei vari modelli è bassa perché ci sono pochi pazienti che rientrano nello stato intermedio e questo non permette al modello di apprendere adeguatamente.

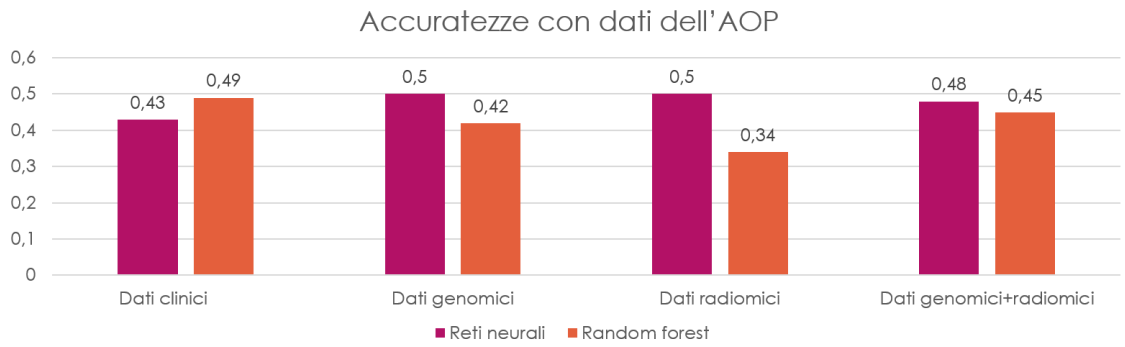

Figura 3.9: Accuratezze con 3 stati

Per ogni stadio del tumore è stata calcolata la distribuzione dei pazienti nelle tre categorie. La Figura 3.10 mostra la stratificazione dei pazienti per stadio.
Percentuali dei pazienti con stadio III:

AND: 74%
DOD: 8.7%
AWD: 17.3%

Percentuali dei pazienti con stadio IVa:

AND: 50%
DOD: 45%
AWD: 5%

Percentuali dei pazienti con stadio IVb:

AND: 33.3%
DOD: 55.6%
AWD: 11.1%

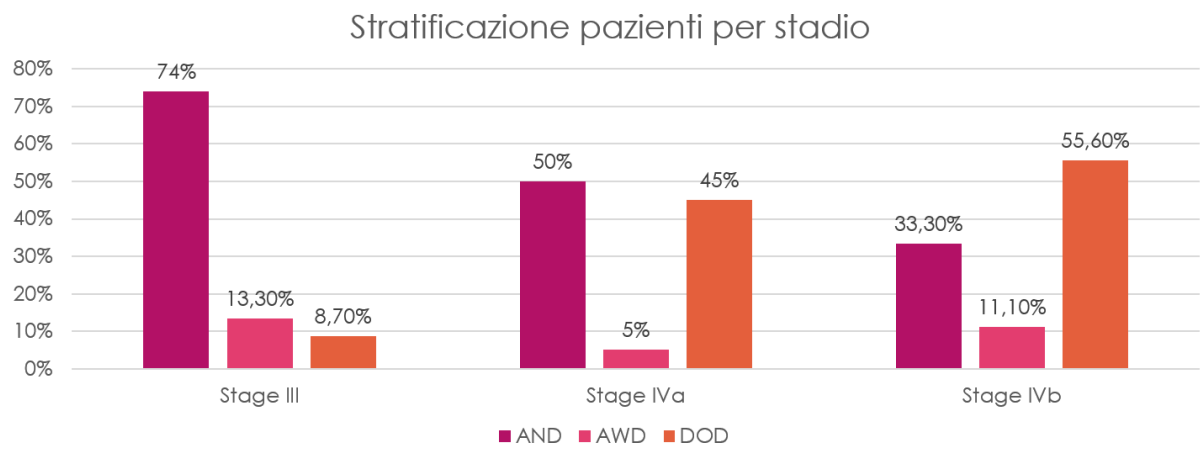

Figura 3.10: Stratificazione dei pazienti per stadio

3.6 Confronto con accuratezza T e stadi

Per vedere quanto i modelli di previsione creati fossero efficaci, è stato effettuato un confronto con la precisione della prognosi del TNM.

Prima è stato necessario definire una metrica comune: il rischio.
Se il paziente è deceduto entro i 5 anni dalla diagnosi significa che il tumore era ad alto rischio, altrimenti a basso rischio.
In seguito è stato suddiviso il parametro T del TNM della settima edizione nelle due categorie.
Se il TNM è T1 o T2, il tumore è considerato a basso rischio, mentre se T è T3, T4a o T4b, il tumore è ad alto rischio.
Per misurare l'accuratezza del T dai dati clinici è stata selezionata la colonna con l'ID del paziente e quella con il T della settima edizione, che ha i dati completi.
Con la funzione *compute_alive_after*, creata in precedenza, è stato realizzato un vettore NumPy con la previsione corretta associando 0, quindi il decesso, al rischio alto e 1, quindi la vita, al rischio basso.
In un altro vettore NumPy è stato sostituito il T con il rischio corrispondente.
La percentuale di accuratezza del T è stata calcolata facendo il valore assoluto della differenza tra i due vettori, così erano settati a 1 gli errori del T. In seguito è stata fatta la somma di questi errori e calcolata la percentuale di accuratezza che risulta essere del

54%, come mostrato in Figura 3.11. L'accuratezza del T è del 54%, anche utilizzando 3 stati e mettendo T3 nel rischio intermedio.

Lo stesso procedimento è stato applicato agli stadi.
Gli stadi I, II, III sono associati al basso rischio e gli stadi IVa, IVb ad alto rischio.
La percentuale calcolata è del 63%.

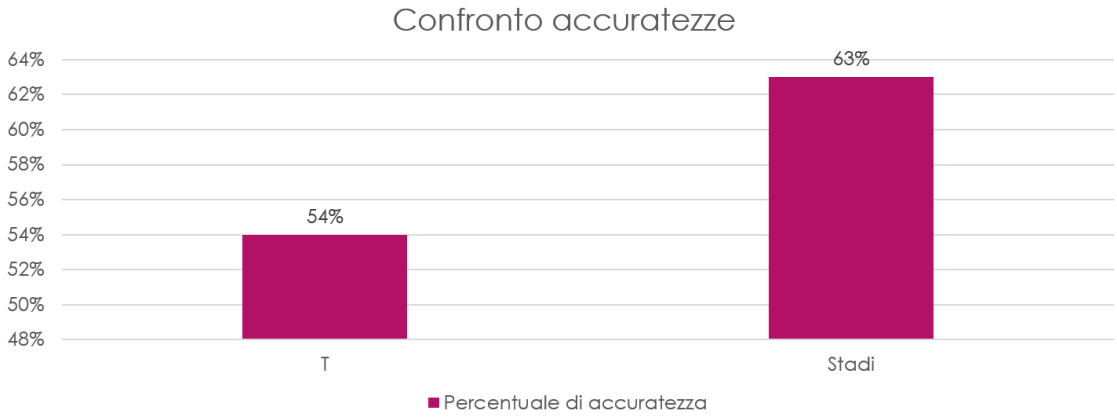

Figura 3.1: Confronto accuratezze T e stadi

4. Conclusioni

Le caratteristiche genomiche scoperte grazie all'utilizzo innovativo di nuovi modelli di analisi statistica (Machine Learning e Big Data) potrebbero offrire il grande vantaggio di evidenziare gli aspetti molecolari riguardanti l'eziologia dei tumori e la trasformazione maligna, consentendo di chiarire le principali vie biomolecolari coinvolte nei processi di oncogenesi.

Per quanto riguarda l'applicazione delle tecniche di Machine Learning, possiamo dire che i risultati ottenuti sono incoraggianti. In futuro sarebbe interessante fornire più dati ai modelli predittivi in modo da consentire loro un sufficiente addestramento, ottenendo così una maggiore accuratezza come è stato dimostrato utilizzando l'intera quantità di dati clinici del progetto europeo e non solo i dati provenienti dall'AOP.

Le precisioni ottenute si basano sull'associazione "basso rischio - paziente vivo 5 anni dopo" e "alto rischio - paziente morto 5 anni dopo". Forse ci sono altre associazioni che potrebbero essere più efficaci e meno rigide.

In tutti i modelli creati le RF si comportano meglio e hanno una precisione maggiore rispetto alla rete neurale utilizzata in questa analisi.

Costruire una RF è molto più semplice perché è necessario solo decidere il numero di alberi e per i parametri rimanenti si possono usare i valori di default. RF inoltre riduce l’overfitting.

Costruire una rete neurale ad alte prestazioni è più complesso. Richiede di specificare l'architettura della rete, il numero di strati e di neuroni per ogni strato, la funzione di attivazione e molti altri parametri, e restituisce una precisione inferiore rispetto alla RF.

È possibile, facendo diverse prove, modificare i parametri e ottenere una maggiore precisione, ma in ogni caso costruire una RF ad alte prestazioni è più facile e semplice.

Quindi per il futuro sarebbe una soluzione migliore utilizzare i modelli RF: non solo in termini di semplicità di utilizzo, ma anche in termini di accuratezza delle previsioni.

L'obiettivo principale sarebbe fornire ai clinici uno strumento diagnostico utile per stratificare meglio gli OSCC, soprattutto quelli in stadio avanzato, rafforzando il concetto di "unicità" e, allo stesso tempo, quello di "variabilità" degli HNC.

Questo è un argomento chiave, poiché una migliore stratificazione dei tumori porta ad un miglioramento del processo decisionale clinico, con conseguente beneficio per il paziente sia dal punto di vista della sopravvivenza che dal punto di vista della qualità della vita.

Riguardo ai dati analizzati, quelli che aprono possibilità di ulteriori analisi e studi sono quelli genomici.

Sebbene alcuni centri, come l'AOP - Dipartimento Maxillo Facciale, abbiano compiuto sforzi per valutare il profilo mutazionale dei tumori HNSCC dei loro pazienti, al giorno d'oggi, i tumori nella maggior parte dei pazienti con HNSCC rimangono non caratterizzati, anche quando questi pazienti presentano una malattia ricorrente o metastatica. Inoltre, il ruolo del carico mutazionale del tumore come biomarcatore predittivo rimane non completamente compreso e sono in corso sforzi per stabilire nuove linee guida.

L'analisi genomica, eventualmente seguita da uno studio trascrittomico, potrebbe essere un nuovo potente strumento per definire diversi programmi di espressione intratumorale e differenze tra tumori di pazienti diversi.

Per una medicina di precisione devono essere indirizzate sia le decisioni cliniche che la terapia sul paziente e sulla sua malattia. Bisogna focalizzarsi maggiormente sull'unicità del cancro in ogni paziente ed evitare di considerare le persone affette come se fossero un unico gruppo omogeneo, aprendo le porte a un nuovo concetto di medicina e terapia fatto su misura del paziente.

Le tecnologie genomiche hanno il potenziale per cambiare le basi della prassi clinica oncologica sia in campo diagnostico che terapeutico. Le informazioni sulla patologia

della malattia dell'HNSCC generate dalle tecnologie genomiche potrebbero essere tradotte in:

a) test diagnostici di biomarcatori di proliferazione cellulare, danno o morte cellulare e metabolismo cellulare

b) prodotti terapeutici attualmente nella prassi clinica per il trattamento dell'HNSCC.

È importante sottolineare che i biomarcatori nei tessuti o nei fluidi corporei identificati dalla genomica offrono un mezzo per rilevare la malattia precoce o ricorrente nei pazienti a rischio di HNSCC. Questa tecnologia ha il potenziale aggiuntivo di caratterizzare gli individui che potrebbero avere una risposta alla chemioterapia e/o alla radioterapia, in modo che il trattamento possa essere adattato al singolo paziente.

Come approccio preventivo, le informazioni basate sul genoma del potenziale rischio di HNSCC possono consentire un intervento precoce con misure preventive, che possono in definitiva ridurre i costi sanitari.

In campo terapeutico, una migliore conoscenza della genetica dei tumori può portare alla scoperta di nuovi bersagli molecolari, con il conseguente sviluppo di nuovi farmaci mirati più efficaci nel trattamento del cancro. In questo senso, sono attualmente in corso nuovi studi per verificare se sia possibile aumentare la sopravvivenza dei pazienti con HNC grazie all'utilizzo di inibitori specifici di geni iperespressi. È quindi essenziale continuare a cercare le mutazioni più frequenti dell'OSCC al fine di migliorare la ricerca e lo sviluppo di nuovi farmaci e terapie mirate.

Inoltre, lo sviluppo di nuovi strumenti prognostici basati sull'analisi genomica offre la possibilità di una stratificazione più precisa al fine di guidare meglio gli approcci terapeutici clinici e chirurgici.

Infine, la combinazione di tecnologie genomiche può consentirci di monitorare un gran numero di percorsi cellulari chiave contemporaneamente e può fornire una migliore comprensione integrata dell'HNSCC.

Bibliografia

Chollet F. Deep Learning with Python. Shelter Island: Manning Publications Co; 2018.

TNM: Brierley J., Gospodarowicz M., Wittekind C. TNM Classification of Malignant Tumours. Eight Edition. Oxford, UK ; Hoboken, NJ :John Wiley & Sons; 2017.

Osazuwa-Peters N, Simpson MC, Zhao L, et al. Suicide risk among cancer survivors: Head and neck versus other cancers. Cancer 2018.

Amin MB. AJCC Cancer Staging Manual. 2017.

Larue RTHM, van de Voorde L, Berbée M, et al. A phase 1 "window-of-opportunity" trial testing evofosfamide (TH-302), a tumour-selective hypoxia-activated cytotoxic prodrug, with preoperative chemoradiotherapy in oesophageal adenocarcinoma patients. BMC Cancer 2016.

Lilloni G., Implementazione del sistema di stadiazione TNM nei tumori testa-collo mediante firma genomica messa a punto in corso del progetto europeo BD2Decide, tesi di laurea in Medicina e Chirurgia, Università degli Studi di Parma, a.a. 2018-2019, relatore Prof. T. Poli.

OMIM: Baltimore MM-NI of GMJHU. Online Mendelian Inheritance in Man. https://www.omim.org/about (accessed June 30, 2021).

Lopez-Perez L, Hernandez L, Ottaviano M, et al. BD2Decide: Big data and models for personalized head and neck cancer decision support. In: Proceedings - IEEE Symposium on Computer-Based Medical Systems. 2019.

Big Data and models for personalized Head and Neck Cancer Decision support - Workshop. Milan, 2018.

Mes SW, te Beest D, Poli T, et al. Prognostic modeling of oral cancer by gene profiles and clinicopathological co-variables. Oncotarget 2017.

National Comprehensive Cancer Network, NCCN Clinical Practice Guidelines in Oncology - Head and Neck Cancers. Version 2.2014, NCCN. 2014.

Python: https://www.python.org/

Keras: https://keras.io/

Tensorflow: https://www.tensorflow.org/

Random Forest: https://www.ibm.com/

Decision Tree: https://careerfoundry.com/

Scikit-learn: https://scikit-learn.org/

NumPy: https://numpy.org/

Pandas: https://pandas.pydata.org/