



Università degli Studi di Cagliari

Corso di laurea in Ingegneria Elettrica e Corso di Laurea in Informatica Applicata e Data Analytics

Corso di Machine Learning e Data Mining

*Sviluppo di modelli di machine learning per la rilevazione di energia
gamma nella bassa atmosfera.*

Docente:

Prof. Lorenzo Putzu

Studenti:

Gabriele Carta

Marco Boi

Alice Mastio

Emanuele Piras

Lorenzo Agostino Cadinu

Sommario

1.	Introduzione	4
	Riferimenti	5
2.	Analisi dei dati	6
2.1	Analisi dei dati – Descrizione del dataset	6
2.2.	Analisi dei dati – Risultati.....	11
2.3.	Analisi dei dati – Discussione.....	15
3.	Classificatori	17
3.1	Classificatore logico: alberi decisionali	18
3.2.	Classificatore a istanze kNN	27
3.3.	Classificatore custom: classificatore multiplo	33
4.	Pre-processing.....	35
4.1.	Confronto Bilanciamento.....	37
4.2.	Confronto features.....	76
5.	Conclusione e scelta del classificatore migliore	94

1. Introduzione

I dati forniti dal dataset sono stati generati nel contesto del Monte Carlo Program (Corsika), descritto in: D. Heck et al., CORSIKA, A Monte Carlo code to simulate extensive air showers, Forschungszentrum Karlsruhe FZKA 6019 (1998) [1]. I dati sono stati simulati per studiare la registrazione di particelle gamma ad alta energia in un telescopio atmosferico Cherenkov installato a terra, utilizzando tecniche di imaging.

I raggi gamma che vagano nello spazio, una volta entrati nell'atmosfera terrestre vengono in minima parte assorbiti dalle particelle che compongono la stessa. Il processo di eccitazione delle particelle negli strati più alti dell'atmosfera genera un'emissione di fotoni. Questi fotoni si sviluppano a cono nella direzione di scontro e si dirigono verso la terra. Il telescopio Cherenkov, per mezzo di una lastra piana sensibile, rileva i raggi gamma ad alta energia sfruttando la radiazione emessa dalle particelle eccitate. Utilizzando delle tecniche di imaging vengono generate delle immagini che consentono di rilevare i raggi gamma provenienti dallo spazio. Tuttavia, i raggi gamma non sono le uniche fonti di particelle energetiche nello spazio, essi vengono anche provocati da interazioni internucleari (interazione forte), le cui particelle sono dette adroni, che per il nostro scopo sono background da eliminare. L'obiettivo della tesina è di distinguere tra i fotoni gamma primari (Classe *g*) e gli adroni (Classe *h*, da hadron) attraverso opportune tecniche di classificazione. Le caratteristiche delle immagini generate dal sistema sono descritte da dieci attributi di dati che permettono la discriminazione tra i due tipi di segnale. I record, ciascuno descritto da dieci attributi, sono raccolti in un dataset raggiungibile nel sito web [2].

Riferimenti

- [1] D. Heck et al., CORSIKA, A Monte Carlo code to simulate extensive air showers, Forschungszentrum Karlsruhe FZKA 6019 (1998).
- [2] <https://archive.ics.uci.edu/dataset/159/magic+gamma+telescope> (accesso 22/01/2024 15:00)

2. Analisi dei dati

Questo capitolo si focalizza sull'analisi dei dati del loro dataset e della loro qualità. La qualità del dataset è valutata in base alla presenza di rumore e outlier.

2.1 Analisi dei dati – Descrizione del dataset

Il dataset è una data matrix di dimensioni 19020x11. La data matrix è costituita da 10 vettori colonna, uno per ogni attributo, e 19020 righe, una per ogni record. Il vettore delle classi è composto da una sola colonna e 19020 righe. Tutti i valori della matrice sono continui e non sono presenti dati mancanti. Una breve rappresentazione è riportata in figura 2.1.1a e 2.1.1b.

Index	fLenath	fWidth	fSize	fConc	fConc1	fAsvm	fM3Lona	fM3Trans	fAlpha	fDist	class
0	28.7967	16.0021	2.6449	0.3918	0.1982	27.7004	22.011	-8.2027	40.092	81.8828	g
1	31.6036	11.7235	2.5185	0.5303	0.3773	26.2722	23.8238	-9.9574	6.3609	205.261	g
2	162.052	136.031	4.0612	0.0374	0.0187	116.741	-64.858	-45.216	76.96	256.788	g
3	23.8172	9.5728	2.3385	0.6147	0.3922	27.2107	-6.4633	-7.1513	10.449	116.737	g
4	75.1362	30.9205	3.1611	0.3168	0.1832	-5.5277	28.5525	21.8393	4.648	356.462	g
5	51.624	21.1502	2.9085	0.242	0.134	50.8761	43.1887	9.8145	3.613	238.098	g
6	48.2468	17.3565	3.0332	0.2529	0.1515	8.573	38.0957	10.5868	4.792	219.087	g
7	26.7897	13.7595	2.5521	0.4236	0.2174	29.6339	20.456	-2.9292	0.812	237.134	g
8	96.2327	46.5165	4.154	0.0779	0.039	110.355	85.0486	43.1844	4.854	248.226	g
9	46.7619	15.1993	2.5786	0.3377	0.1913	24.7548	43.8771	-6.6812	7.875	102.251	g
10	62.7766	29.9104	3.3331	0.2475	0.1261	-33.9065	57.5848	23.771	9.9144	323.094	g
11	18.8562	16.46	2.4385	0.5282	0.2933	25.1269	-6.5401	-16.9327	11.461	162.848	g

Figura 2.1.1b Rappresentazione di una prima porzione del dataset

Index	fLenath	fWidth	fSize	fConc	fConc1	fAsym	fM3Lona	fM3Trans	fAlpha	fDist	class
13005	25.9859	23.1756	2.3015	0.5719	0.3495	-15.3044	-6.6741	12.9296	6.8391	119.988	h
13006	81.0314	9.8854	3.0858	0.3233	0.2138	57.4644	60.1031	8.0262	21.007	180.324	h
13007	46.9654	11.1482	2.8176	0.3839	0.2009	40.3412	23.1535	5.657	26.0156	66.4015	h
13008	25.4615	3.8234	2.4885	0.6954	0.3595	19.4852	1.3408	5.5778	2.1818	170.596	h
13009	128.652	62.179	3.5485	0.1202	0.068	-12.2082	80.0568	53.1531	87.534	232.057	h
13010	13.9498	11.2494	2.4374	0.7756	0.496	-1.2467	-6.5713	-9.7086	30.9788	228.609	h
13011	25.2651	17.9293	2.8822	0.3446	0.1836	-18.3011	24.4749	-7.6379	46.9569	87.8103	h
13012	120.922	71.0392	3.6757	0.2054	0.0901	-58.0453	-141.16	45.3024	21.4663	399.041	h
13013	45.0726	13.8787	2.8026	0.4685	0.2339	8.6753	14.8679	-3.1617	39.4333	178.279	h
13014	19.3677	16.6452	2.42	0.4411	0.2338	1.398	-8.2798	7.2769	41.666	213.354	h
13015	26.1415	12.8662	2.679	0.4377	0.2251	29.836	18.0079	-7.718	68.852	216.686	h
13016	31.4245	15.3624	2.5966	0.3949	0.257	9.2765	-19.2612	9.6225	33.559	218.101	h

Figura 2.1.1b Rappresentazione di una seconda porzione del dataset

I dieci attributi del dataset sono numerici di tipo *float* e continui. Inoltre, i dati degli attributi della data matrix sono tutti categorizzati come *ratio* in quanto su questi dati numerici sono ammesse tutte le quattro operazioni matematiche (somma, sottrazione, moltiplicazione, divisione), sono distinguibili (ammessa operazione di confronto $=, \neq$) e sono ordinabili ($\leq, <, >, \geq$). Il vettore della label di classe y invece, è binario in quanto le classi possibili sono g (segna di raggi γ) o h (segna di sottofondo). Il significato di ogni attributo è riportato nella tabella 2.1.

<i>Variable Name</i>	<i>Role</i>	<i>Type</i>	<i>Description</i>	<i>Units</i>	<i>Missing Values</i>
fLength	Feature	Continuous	major axis of ellipse	mm	no
fWidth	Feature	Continuous	minor axis of ellipse	mm	no
fSize	Feature	Continuous	10-log of sum of content of all pixels	#phot	no
fConc	Feature	Continuous	ratio of sum of two highest pixels over fSize		no
fConc1	Feature	Continuous	ratio of highest pixel over fSize		no
fAsym	Feature	Continuous	distance from highest pixel to center, projected onto major axis		no
fM3Long	Feature	Continuous	3rd root of third moment along major axis	mm	no
fM3Trans	Feature	Continuous	3rd root of third moment along minor axis	mm	no
fAlpha	Feature	Continuous	angle of major axis with vector to origin	deg	no
fDist	Feature	Continuous	distance from origin to center of ellipse	mm	no

Tabella 2.1.1 Descrizione degli attributi del dataset

Dalla prima analisi del vettore della label di classe emerge che il numero di record appartenenti alla classe *g* è 12332, mentre i record della classe *h* sono 6688. Pertanto, il 64.84 % dei record è della classe *g* e il 35.16% è della classe *h*.

Si può affermare che si è in presenza di un dataset leggermente sbilanciato. Un dataset si dice “non bilanciato” quando esiste una differenza significativa nel numero di record appartenenti a una classe rispetto all’altra. È possibile che lo squilibrio negli esempi tra le classi sia stato causato dal modo in

cui gli esempi sono stati raccolti o campionati, poiché sappiamo che il numero di record appartenenti alla classe h è molto più alto del numero di record della classe g nella realtà. Per convenzione si può identificare la classe g (di interesse) con la classe **positiva** e la classe h con la classe **negativa**. È richiesto idealmente che il classificatore sia in grado di classificare correttamente tutti i record della classe h , cioè la classe negativa. Infatti, se il classificatore assegna erroneamente a un record in realtà h la label di classe g è un grave errore, in quanto si sta affermando che un segnale di fondo h è un segnale di interesse g . È più tollerabile assegnare erroneamente un record della classe g alla classe h , cioè è più accettabile classificare un segnale di interesse (segna g) come un segnale di fondo (segna h). Pertanto, il classificare deve essere molto specifico per la classe negativa. Idealmente deve essere in grado di identificare tutti i negativi correttamente ed è accettabile classificare qualche positivo come negativo. Dunque, si desidera che il classificare sia caratterizzato da un elevato *TNR* (specificità). Un classificatore molto specifico classifica correttamente tutti i negativi come negativi (*True Negative*), e ogni tanto classifica erroneamente qualche positivo come negativo (*FP*, *False Positive*). Ai fini della valutazione del modello di classificazione si utilizza la *precision*. La *precision* è definita così:

$$p = \frac{TP}{TP + FP} \quad (1)$$

La *precision* ha il valore massimo quando il numero di falsi positivi che tende a zero. Siccome l'obiettivo del classificatore è la minimizzazione del numero di record h classificati come g cioè il numero di falsi positivi, si deve diminuire il numero di *FP* il più possibile. Per questo motivo la metrica *precision* può essere una buona scelta.

L'analisi dei dati è stata eseguita principalmente tramite una funzione di python chiamata “*dataAnalysis()*” costruita appositamente, presente nel file *analisi_dati.py*. Questa funzione prende in ingresso il dataset degli attributi e calcola per ogni vettore attributo moda, media aritmetica, varianza, deviazione standard e calcola i coefficienti di Pearson tra tutti gli attributi.

All'interno dello stesso file, sono presenti altre quattro funzioni: “*create_hist()*” e “*check_duplicates()*”, “*create_pie_class_distribution()*” e *print_analysi()*. La funzione “*create_hist()*” prende in ingresso il dataset e genera un subplot con 10 istogrammi, ciascuno mostrante la distribuzione della singola feature in base alla classe di appartenenza. La funzione “*check_duplicates()*” riceve il dataset come argomento, verifica quanti valori unici sono presenti per ogni feature e crea un istogramma a barre. La funzione *create_pie_class_distribution()* riceve il dataset come argomento e genera un grafico a torta per rappresentare la percentuale di classi presente nel database. La funzione *print_analysi()* riceve il dataset come argomento e stampa a video la media,

moda, mediana, deviazione standard, varianza, range e i percentili 10%, 25%, 50%, 75%, 90% di ogni attributo. Queste metriche sono essenziali per analizzare la qualità dei dati.

Attraverso gli istogrammi, si possono identificare gli attributi potenzialmente migliori, oltre a quelli che non potrebbero contribuire in maniera significativa. Ad esempio si osserva che gli attributi *fSize*, *fConc*, e *fConc1* sono formati da pochi valori unici e distribuiti in modo simile, perciò è improbabile riuscire a trovare una netta distinzione tra i record appartenenti alla classe “gamma” e “hadron” utilizzando soltanto questi attributi.

2.2. Analisi dei dati – Risultati

L’output della funzione *dataAnalysis()* è costituito dalla media campionaria, il massimo e minimo valore, il range, mediana, i percentili 10%, 25%, 50%, 75%, 90%, la varianza campionaria, deviazione standard campionaria, la moda e il numero di volte in cui si ripete la moda. Questi dati sono calcolati per ogni attributo. I risultati sono disponibili nella tabella 2.2.1.

	fLength	fWidth	fSize	fConc	fConc1	fAsym	fM3Long	fM3Trans	fAlpha	fDist
<i>Numero di record</i>	19020	19020	19020	19020	19020	19020	19020	19020	19020	19020
<i>Valore massimo attributo</i>	334.177	256.382	5.3233	0.893	0.6752	575.2407	238.321	179.851	90	495.561
<i>Media aritmetica</i>	53.25015	22.18097	2.825017	0.380327	0.214657	-4.33175	10.54554	0.249	27.645	193.818
<i>Valore minimo attributo</i>	4.2835	0	1.9413	0.0131	0.0003	-457.916	-331.78	-205.895	0	1.2826
<i>Range = max - min</i>	329.894	256.382	3.382	0.8799	0.6749	1033.16	570.101	385.746	90	494.278
<i>Deviazione standard campionaria</i>	42.36485	18.34606	0.472599	0.182813	0.110511	59.20606	51	20.827	26.103	74.731
<i>Varianza campionaria</i>	1794.78	336.578	0.223349	0.033421	0.012213	3505.36	2601.01	433.782	681.399	5584.84
<i>Mediana</i>	37.1477	17.1399	2.7396	0.35415	0.1965	4.01305	15.3141	0.666	17.679	191.851
<i>Moda</i>	12.9176	0	2.1508	0.6	0.194	0	0	0	0.0002	100.395
<i>Numeri volte ripetizione della moda</i>	3	98	27	16	18	41	39	59	7	3
<i>Percentile 10%</i>	19.150	9.6083	2.286	0.162	0.0873	-70.212	-34.857	-18.011	1.9326	96.5821
<i>Percentile 25%</i>	24.336	11.863	2.477	0.235	0.1284	-20.586	-12.842	-10.8	5.547	142.492
<i>Percentile 50%</i>	37.1477	17.139	2.739	0.354	0.196	4.013	15.314	0.666	17.679	191.851
<i>Percentile 75%</i>	70.1222	24.7395	3.1016	0.5037	0.285225	24.0637	35.8378	10.9464	45.8835	240.564
<i>Percentile 90%</i>	105.459	38.6731	3.47761	0.64781	0.37512	46.8335	67.024	18.314	70.9384	292.238

Tabella 2.2.1 Caratteristiche degli attributi

La stessa funzione restituisce la matrice della correlazione lineare di Pearson. Questa matrice di correlazione ha come ingressi i coefficienti di Pearson che permettono di rilevare un eventuale presenza di una correlazione lineare tra gli attributi. Al contrario, non è possibile rilevare la correlazione non lineare tra gli attributi utilizzando questa matrice.

	<i>fLength</i>	<i>fWidth</i>	<i>fSize</i>	<i>fConc</i>	<i>fConc1</i>	<i>fAsym</i>	<i>fM3Long</i>	<i>fM3Trans</i>	<i>fAlpha</i>	<i>fDist</i>
<i>fLength</i>	1	0.770512	0.702454	-0.631	-0.59815	-0.36856	-0.11975	0.013389	-0.00878	0.418466
<i>fWidth</i>	0.770512	1	0.717517	-0.60978	-0.58114	-0.26696	-0.17623	0.039744	0.066061	0.336816
<i>fSize</i>	0.702454	0.717517	1	-0.85085	-0.80884	-0.15986	0.095157	0.015455	-0.18668	0.437041
<i>fConc</i>	-0.631	-0.60978	-0.85085	1	0.976412	0.112272	-0.1219	-0.01129	0.235272	-0.32833
<i>fConc1</i>	-0.59815	-0.58114	-0.80884	0.976412	1	0.100159	-0.11877	-0.01097	0.229799	-0.30462
<i>fAsym</i>	-0.36856	-0.26696	-0.15986	0.112272	0.100159	1	0.274045	0.002553	-0.05569	-0.20673
<i>fM3Long</i>	-0.11975	-0.17623	0.095157	-0.1219	-0.11877	0.274045	1	-0.0172	-0.18627	0.037025
<i>fM3Trans</i>	0.013389	0.039744	0.015455	-0.01129	-0.01097	0.002553	-0.0172	1	0.004659	0.011427
<i>fAlpha</i>	-0.00878	0.066061	-0.18668	0.235272	0.229799	-0.05569	-0.18627	0.004659	1	-0.22056
<i>fDist</i>	0.418466	0.336816	0.437041	-0.32833	-0.30462	-0.20673	0.037025	0.011427	-0.22056	1

Tabella 2.2.2 Matrice di correlazione di Pearson

L'output della funzione *create_boxplot(dataset)* sono una serie box plot dei dieci attributi, riportato nella figura 2.2.1 in cui è possibile osservare la distribuzione dei dati. Un box plot mostra la distribuzione dei dati per una variabile continua e consente di visualizzare il centro e la distribuzione dei dati. Inoltre, lo si può usare come strumento visivo per la verifica della normalità o per identificare possibili outlier. La linea centrale nella scatola rappresenta la mediana dei dati. La metà dei dati si trova sopra questo valore, l'altra metà sotto. Se i dati sono simmetrici, la mediana è al centro della scatola. Se, invece, i dati sono asimmetrici, la mediana sarà più vicina alla parte superiore o a quella inferiore della scatola. La parte inferiore e superiore della scatola mostrano il 25 % e il 75 % quantile, o percentile. Questi due quantili sono chiamati anche quartili, poiché ciascuno di essi esclude un quarto (25 %) dei dati. La lunghezza della scatola è la differenza tra i due percentili e si chiama range interquartile (IQR). Le linee che si estendono a partire dalla scatola sono chiamate baffi. I baffi rappresentano la variazione dei dati attesa e si estendono per 1.5 volte dall'IQR dalla parte superiore e inferiore della scatola. Se i dati non arrivano fino alla fine dei baffi, significa che i baffi si estendono fino ai valori di dati minimi e massimi. Se, invece, i dati ricadono sopra o sotto la fine dei baffi, sono rappresentati come punti, denominati spesso outlier. Un outlier è più estremo della variazione attesa. Vale la pena esaminare questi punti di dati per determinare se sono errori o outlier. I baffi non comprendono gli outlier.

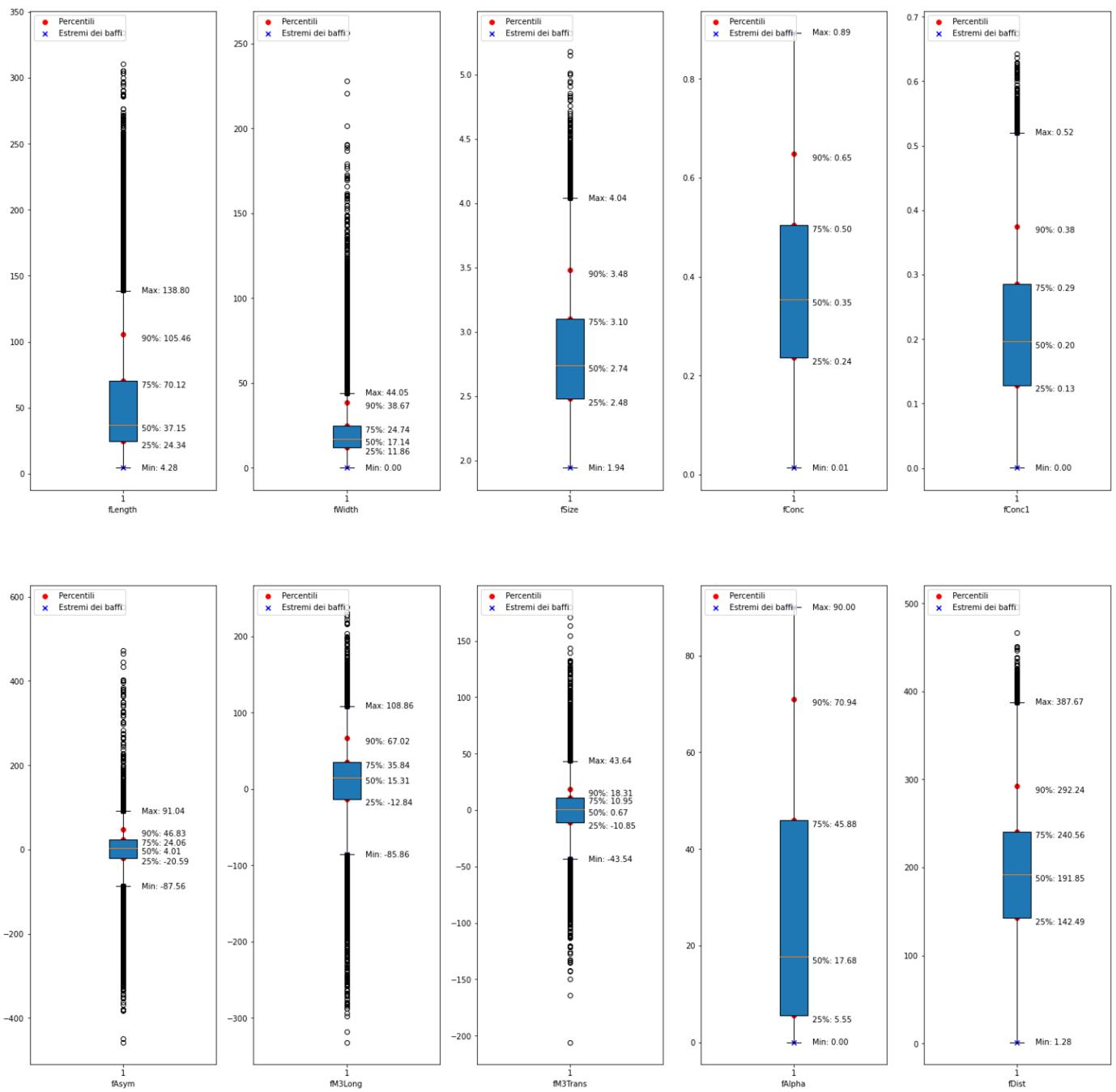


Figura 2.2.1 Box plot degli attributi

La funzione `create_hist()` riceve in ingresso gli attributi e genera per ogni attributo un istogramma a barre. Questo tipo di istogramma è utile per i dati continui in quanto dà un'idea della distribuzione dei dati. Nella figura 2.2.2 sono stati riportati gli istogrammi per ogni attributo.

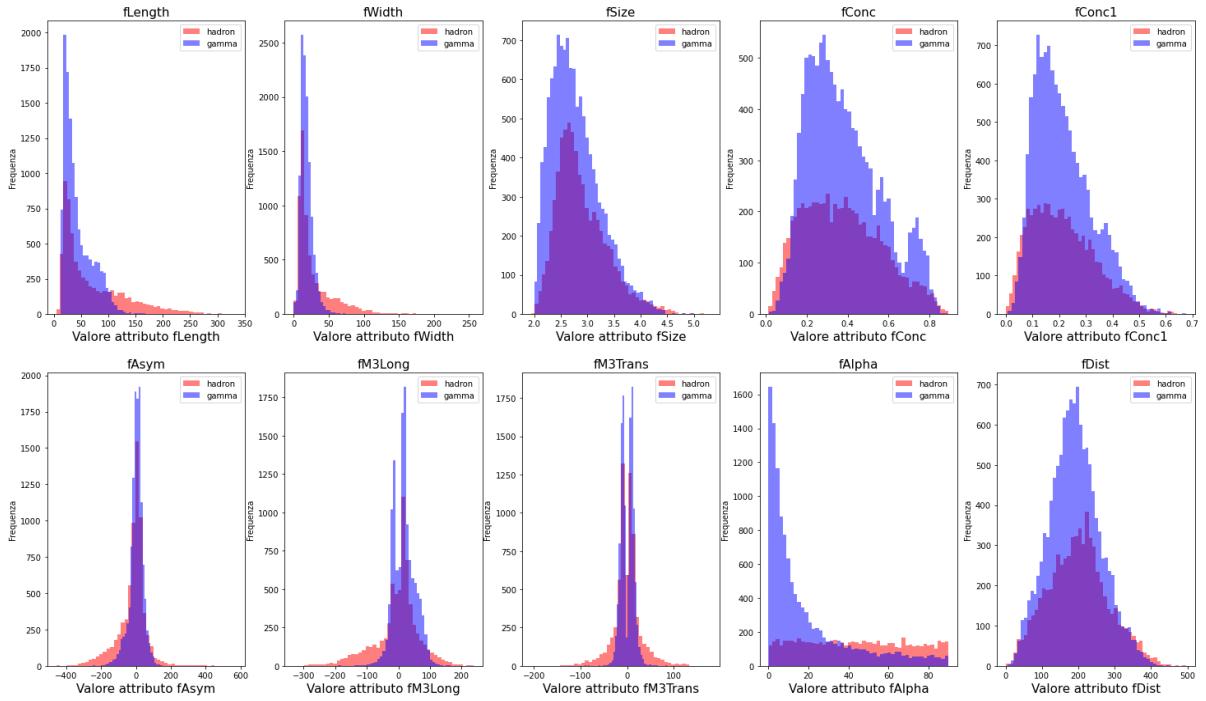


Figura 2.2.2 Istogrammi degli attributi

Infine, le ultime funzioni del file analisi_dati.py sono la `check_duplicates()` che riceve in input gli attributi e restituisce un grafico a barre in cui vengono rappresentati i valori unici degli attributi. La figura 2.2.3 serve a quantificare quanti sono i dati unici, ossia senza duplicati. Se un attributo ha un valore basso vuol dire che presenta molti dati duplicati. Questo grafico è utile assieme all’analisi della moda in quanto permette di capire l’importanza di un attributo. Infine, la funzione `create_pie_class_distribution()` riceve in ingresso il dataset degli attributi e delle classi e mostra un grafico a torta su come sono distribuite le due classi. Il grafico a torta è stato presentato nel paragrafo precedente (vedi figura 2.1.2).

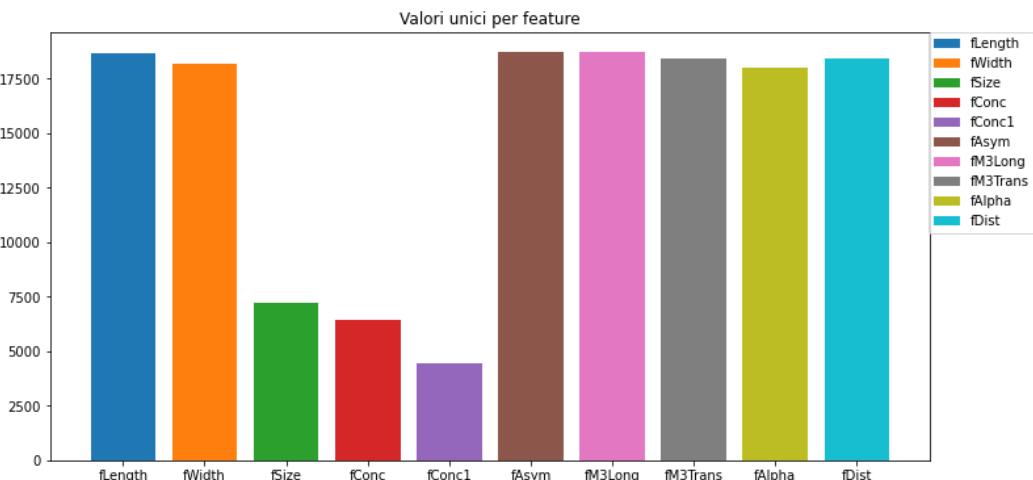


Figura 2.2.3 Rappresentazione dei valori unici negli attributi

2.3. Analisi dei dati – Discussione

Gli attributi presentano caratteristiche eterogenee. Alcuni attributi come *fSize*, *fConc*, *fAsym* hanno una dispersione dei dati contenuta. Infatti, i loro valori di varianza e deviazione standard campionari sono relativamente piccoli, pari a 0.011 e 0.182 per l’attributo *fConc1* e *fConc* rispettivamente. Gli altri sette attributi invece presentano un’elevata dispersione nei dati sperimentali, la maggiore è presente nell’attributo *fDist*, con una deviazione standard di 74.73 mm e una varianza campionaria di 5584.84 mm². Comparabili dispersioni nei dati sono presenti negli attributi *fAsym* e *fFM3long* con deviazioni standard 59.2 e 51 rispettivamente. I valori di range offrono le medesime informazioni. Una comprensione della dispersione e quindi della qualità dei dati si coglie meglio nello studio dei box plot per ogni attributo. Come ci si aspetta dall’osservazione delle medie, mediane e della deviazione standard e varianza, il box plot dell’attributo *fConc* non presenta alcun outlier. Infatti, gli outlier hanno un effetto importante sulla media, deviazione standard e varianza mentre non hanno effetto sulla mediana. *fConc* è l’unico attributo del dataset in cui la media e la mediana sono molto vicine. Questo è dovuto al fatto che non ci sono outlier e che i dati sono molto vicini tra di loro. Questo viene confermato anche dal box plot. Anche l’attributo *fAlpha* non presenta outlier ma la sua media si discosta dalla mediana a causa di maggior dispersione presente tra i dati sperimentali. Probabilmente questa caratteristica è affetta da rumore. Questi attributi sono affetti da molto rumore e alcuni outlier. Infatti, si può osservare che la media e la mediana in ogni attributo si discostano sensibilmente. Nel dataset il rumore pare essere la caratteristica principale in quanto un outlier ha la caratteristica di essere un record isolato e ben lontano dagli altri record. In questo dataset si osserva che i record degli attributi tendono a rimanere aggregati tra di loro anche se si distanziano per effetto del rumore. Ci sono alcuni outlier che andrebbero presi in considerazione nello sviluppo dei modelli di classificazione. Il rumore è una caratteristica inevitabile nei sistemi di misura fisici. La rilevazione di particolari raggi gamma è una casistica in cui la strumentazione di misura presenta rumore da molteplici sorgente ed è un fattore da prendere in considerazione. Il rumore e gli outlier devono essere trattati con attenzione in quanto sono fattori che contribuiscono a degradare la qualità di un dataset. La diversificazione degli attributi si riflette anche nella tipologia delle distribuzioni dei dati. Alcuni

attributi hanno distribuzioni che ricordano le distribuzioni di probabilità di Weibull come quelle di *fLength*, *fAlpha*, *fWidth* e *fConc1*. In generale, nessuna di queste ha una distribuzione di tipo gaussiana, se non forse quella di *fAsym* che ricorda una distribuzione a campana molto stretta. Inoltre, i dati degli attributi sono distribuiti in maniera abbastanza asimmetrica. Un aspetto importante da osservare è la presenza di duplicati. Dato che si tratta di dati continui la presenza di duplicati è meno probabile rispetto ai dati discreti. Osservando la figura 2.2.3 si osserva che tutti gli attributi tranne *fSize*, *fConc* e *fConc1* hanno molti dati unici, con pochi duplicati. L'attributo *fConc1* pare avere un grande numero di dati duplicati. Tuttavia, la sua moda è 0.194 e viene ripetuta soltanto 18 volte. Gli attributi *fSize*, *fConc* e *fConc1* hanno una bassa variabilità con molti dati ripetuti. Questi tre attributi potrebbero essere meno significativi per la creazione del modello di classificazione. Al contrario, gli altri sette attributi presentano pochi duplicati e possono essere più significativi. Per concludere è stata calcolata la matrice di correlazione lineare. I coefficienti di questa matrice dicono che alcuni attributi come *fLength*, *fWidth*, *fSize*, *fConc* e *fConc1* sono linearmente correlati tra di loro. Gli altri attributi non sono linearmente correlati, ma è possibile che siano correlati non linearmente. Infatti, gli attributi fM3Long e fM3Trans sono correlati non linearmente con *fWidth* e con *fLength*.

In sintesi, si osserva che il dataset è leggermente sbilanciato, con una maggioranza della classe g. Tutti gli attributi sono affetti da rumore e la maggior parte di essi presentano outlier che vanno inclusi nello studio. Gli attributi *fConc* e *fConc1* presentano la stessa informazione pertanto sono quelli meno significativi nel dataset, al contrario degli altri. Le diverse caratteristiche degli attributi si riflettono anche sulle differenti distribuzioni dei dati degli attributi, asimmetriche e non gaussiane. Sono presenti sia correlazioni lineari che non lineari tra gli attributi.

3. Classificatori

In questo capitolo sono riportati i risultati dei classificatori utilizzati. È stato scelto di utilizzare due classificatori: l'albero decisionale e il k-NN (Nearest Neighbor). La prima tecnica di classificazione testata è albero decisionale offerto dalla libreria di python Scikit Learn. Come metrica di valutazione è stata utilizzato l'indice GINI come misura di impurità. L'indice di GINI è la scelta di default. La seconda funzione di Scikit Learn è il k-NN. Il metodo di Naive Bayes che prevede di ignorare la dipendenza condizionata tra attributi in fase di induzione non è stato utilizzato in quanto non si hanno sufficienti conoscenza per stabilire se c'è una dipendenza tra gli attributi e in quale entità.

Si osserveranno le prestazioni di due modelli: uno logico e l'altro basato sulle istanze. Entrambi i modelli si prestano a un tuning degli vari iperparametri che verrà effettuato sui dati grezzi e sui dati pre-processati, dopo uno split del train test con l'holdout 25% per il test set e il 75% training set. Ognuno dei modelli funziona in base alle caratteristiche dei dati pre-processati, anche se queste regole possono valere in generale per qualsiasi modello. In generale, il k-NN per esempio è più significativo con un numero ridotto di attributi, si presta bene quindi a una selezione o accorpamento di features e ha bisogno di dati standardizzati tra loro; un albero decisionale, invece, può essere più sensibile a dei valori mancanti e ad un dataset sbilanciato.

Nota per le metriche utilizzate:

Come visto nell'analisi dei dati la classe positiva g è in realtà molto più rara rispetto ai numeri che troviamo nel dataset. È importante che essa venga classificata con certezza dal nostro modello di classificazione, pertanto si è deciso di optare per un modello altamente specifico, per evitare che gli scienziati si mettano a studiare dei dati in realtà falsi. La metrica scelta per la valutazione degli iperparametri sarà la massimizzazione della precision (1), ovverosia fare in modo che i classificati come g (positivi) siano il più possibile corretti. Oltre alla *precision* saranno mostrati anche i valori di *accuratezza*, *error rate*, *TP*, *TPR*, *TNR*, *FPR*, *FNR*, *recall*, e *F1* classificati.

3.1 Classificatore logico: alberi decisionali

Gli iperparametri che verranno valutati per gli alberi decisionali sono tre: **la profondità dell’albero, il massimo numero di foglie ed il minimo guadagno** necessario per creare una nuova foglia. Questo terzo iperparametro è leggermente diverso dai primi due: la profondità e il numero di foglie sono assoluti e descrivono il dataset a prescindere dal pre-processing dei dati presentati. Il valore del guadagno minimo invece può presentare alberi decisionali di diverse profondità e numero di foglie a seconda della bontà e delle caratteristiche dei dati che andranno ad indurre il modello. Verrà utilizzato principalmente per scegliere i due iperparametri assoluti in modo da tener conto anche della complessità computazionale e generale del modello. Per esempio, se x è la profondità migliore per le prestazioni dell’albero, ma senza una differenza di guadagno significativa, è possibile trovare una y molto minore, verrà sempre scelta quest’ultima come valore finale.

Per fare il tuning del primo iperparametro viene utilizzata la funzione custom “**dtree_tuning_max_depth()**” che riceve in input gli insiemi di training set e test set e un valore **max_v** che rappresenta il massimo valore da provare per il tuning dell’iperparametro. La funzione itera con un ciclo for diversi alberi decisionali. Restituisce sia una tabella con le metriche e un grafico che paragona la precisione (nell’asse y) al dato iperparametro (nell’asse x), sia un valore chiamato **depthtuned**, che rappresenta il valore di profondità che porta alla massima precisione possibile.

Nel seguente grafico 3.1.1 e nella figura 3.1.2 sono mostrate le metriche per ogni valore di max depth, da 1 a 30. Ovviamente, la profondità massima non può essere zero.

La funzione dice che la profondità di albero che porta precisione migliore è 12, classificando 2824 TP. Si arriva ad una precisione di 0.86.

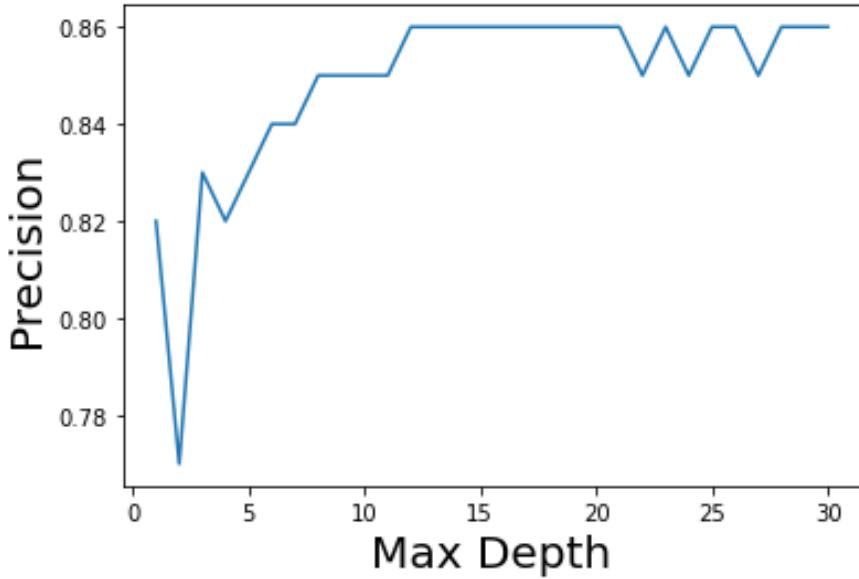


Figura 3.1.1 Rappresentazione della precisione in funzione della massima profondità dell'albero

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Accuracy	0.74	0.79	0.80	0.82	0.83	0.85	0.85	0.84	0.85	0.85	0.85	0.85	0.84	0.84
Error rate	0.26	0.21	0.20	0.18	0.17	0.15	0.15	0.16	0.15	0.15	0.15	0.15	0.16	0.16
TP	2392.00	2963.00	2689.00	2895.00	2858.00	2888.00	2893.00	2841.00	2844.00	2853.00	2838.00	2824.00	2799.00	2784.00
TPR	0.78	0.96	0.87	0.94	0.93	0.94	0.94	0.92	0.92	0.93	0.92	0.92	0.91	0.90
TNR	0.68	0.48	0.67	0.61	0.66	0.68	0.68	0.70	0.70	0.71	0.71	0.72	0.72	0.72
FPR	0.32	0.52	0.33	0.39	0.34	0.32	0.32	0.30	0.30	0.29	0.29	0.28	0.28	0.28
FNR	0.22	0.04	0.13	0.06	0.07	0.06	0.06	0.08	0.08	0.07	0.08	0.08	0.09	0.10
Precision	0.82	0.77	0.83	0.82	0.83	0.84	0.84	0.85	0.85	0.85	0.85	0.86	0.86	0.86
Recall	0.78	0.96	0.87	0.94	0.93	0.94	0.94	0.92	0.92	0.93	0.92	0.92	0.91	0.90
F1	0.80	0.86	0.85	0.87	0.88	0.89	0.89	0.88	0.89	0.89	0.89	0.89	0.88	0.88
	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Accuracy	0.84	0.84	0.83	0.83	0.83	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.81	0.81
Error rate	0.16	0.16	0.17	0.17	0.17	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.19	0.19
TP	2781.00	2766.00	2749.00	2714.00	2707.00	2688.00	2689.00	2678.00	2656.00	2656.00	2672.00	2647.00	2639.00	2636.00
TPR	0.90	0.90	0.89	0.88	0.88	0.87	0.87	0.87	0.86	0.86	0.87	0.86	0.86	0.86
TNR	0.72	0.72	0.72	0.73	0.73	0.73	0.73	0.72	0.74	0.73	0.74	0.74	0.73	0.74
FPR	0.28	0.28	0.28	0.27	0.27	0.27	0.27	0.28	0.26	0.27	0.26	0.26	0.27	0.26
FNR	0.10	0.10	0.11	0.12	0.12	0.13	0.13	0.13	0.14	0.14	0.13	0.14	0.14	0.14
Precision	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.85	0.86	0.85	0.86	0.86	0.85	0.86
Recall	0.90	0.90	0.89	0.88	0.88	0.87	0.87	0.87	0.86	0.86	0.87	0.86	0.86	0.86
F1	0.88	0.88	0.87	0.87	0.87	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86

Figura 3.1.2 Rappresentazione delle metriche di valutazione per i primi 14 valori della massima profondità dell'albero

Per il tuning del secondo iperparametro viene invece utilizzata una seconda funzione custom chiamata “**dtree_tuning_max_leaf_nodes()**” che è esattamente analoga alla prima tranne per l’iperparametro modificato.

Nel grafico 3.1.3 e nella figura 3.1.4 sono rappresentate le metriche per ogni valore di max leaf nodes fino a 30. In questo caso, è necessario partire da un minimo di 2.

La funzione dice che il numero di foglie che porta la precisione migliore è 16, classificando correttamente 2796 record (i True Positive sono 2796). Si arriva ad una precisione di 0.85.

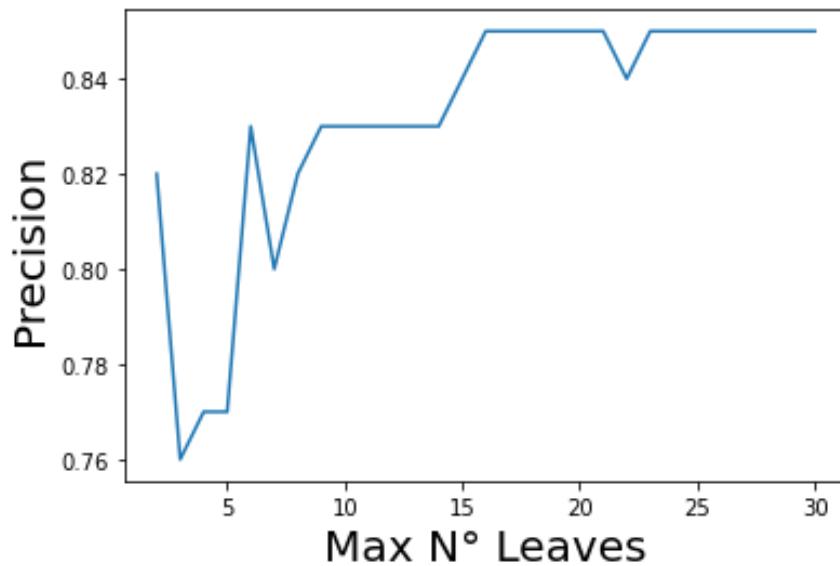


Figura 3.1.3 Rappresentazione delle metriche di valutazione in funzione del numero di foglie

	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Accuracy	0.74	0.78	0.79	0.79	0.80	0.82	0.82	0.83	0.83	0.83	0.83	0.83	0.83	0.83
Error rate	0.26	0.22	0.21	0.21	0.20	0.18	0.18	0.17	0.17	0.17	0.17	0.17	0.17	0.17
TP	2392.00	2984.00	2963.00	2963.00	2689.00	2930.00	2885.00	2831.00	2831.00	2831.00	2831.00	2823.00	2823.00	2813.00
TPR	0.78	0.97	0.96	0.96	0.87	0.95	0.94	0.92	0.92	0.92	0.92	0.92	0.92	0.91
TNR	0.68	0.43	0.48	0.48	0.67	0.57	0.61	0.66	0.66	0.66	0.66	0.66	0.66	0.68
FPR	0.32	0.57	0.52	0.52	0.33	0.43	0.39	0.34	0.34	0.34	0.34	0.34	0.34	0.32
FNR	0.22	0.03	0.04	0.04	0.13	0.05	0.06	0.08	0.08	0.08	0.08	0.08	0.08	0.09
Precision	0.82	0.76	0.77	0.77	0.83	0.80	0.82	0.83	0.83	0.83	0.83	0.83	0.83	0.84
Recall	0.78	0.97	0.96	0.96	0.87	0.95	0.94	0.92	0.92	0.92	0.92	0.92	0.92	0.91
F1	0.80	0.85	0.86	0.86	0.85	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.88
	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Accuracy	0.83	0.83	0.83	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84
Error rate	0.17	0.17	0.17	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
TP	2796.00	2796.00	2783.00	2819.00	2819.00	2819.00	2835.00	2835.00	2835.00	2835.00	2833.00	2828.00	2828.00	2832.00
TPR	0.91	0.91	0.90	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
TNR	0.70	0.70	0.71	0.69	0.69	0.69	0.68	0.69	0.69	0.69	0.70	0.70	0.70	0.70
FPR	0.30	0.30	0.29	0.31	0.31	0.31	0.32	0.31	0.31	0.31	0.30	0.30	0.30	0.30
FNR	0.09	0.09	0.10	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
Precision	0.85	0.85	0.85	0.85	0.85	0.85	0.84	0.85	0.85	0.85	0.85	0.85	0.85	0.85
Recall	0.91	0.91	0.90	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
F1	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88

Figura 3.1.4 Rappresentazione delle metriche di valutazione per le prime 15 foglie

Con la funzione custom “`dtree_tuned()`” è possibile passare i valori dei due iperparametri decisi dalle funzioni precedenti ed indurre un albero decisionale tuned. Viene stampata una tabella con le metriche e lo schema grafico dell’albero decisionale. La precisione ed il numero di *TP* classificati combinando i due iperparametri non differiscono dal modello ottenuto utilizzando solo il numero massimo di foglie, questo perché l’algoritmo si ferma al raggiungere di 12 nodi foglia quando la profondità è ancora solo 7. L’albero a profondità 7 con 16 foglie è rappresentato nella figura 3.1.5 e nella figura 3.1.6 le sue prestazioni.



Figura 3.1.5 Rappresentazione dell'albero decisionale a 16 foglie e profondità 7

	Tuned
Accuracy	0.83
Error rate	0.17
TP	2796.00
TPR	0.91
TNR	0.70
FPR	0.30
FNR	0.09
Precision	0.85
Recall	0.91
F1	0.88

Figura 3.1.6 Prestazioni dell'albero decisionale a 16 foglie e profondità 7

L’albero tuned si mostra complicato e vasto, pertanto è stata creata una funzione custom che si occupa del terzo iperparametro che si è scelto di analizzare: il gain minimo necessario per creare un nuovo nodo. L’interesse è cercare di capire se è possibile ottenere prestazioni paragonabili con strutture di albero nettamente più semplici di quella che massimizza la *precision*. La semplicità del modello non è da trascurare rispetto alle sue prestazioni. L’analisi è condotta attraverso la funzione custom “**dtree_min_gain()**” che accetta gli insiemi di training e test e una lista **l_min_gain** dove è possibile inserire n valori di min impurity decrease con cui creare e addestrare un classificatore. Min impurity decrease rappresenta la differenza minima che deve avere un nodo figlio per essere generato, ossia il guadagno minimo necessario rispetto al nodo parent per ogni nuovo nodo figlio. Per ogni valore viene stampato lo schema grafico dell’albero e una volta terminato il ciclo viene restituita una tabella con le metriche per ogni valore scelto.

Sono stati scelti a livello esplicativo i valori di guadagno minimo: **0.1, 0.05, 0.03, 0.008, 0.0075, 0.005**. Si inizia un guadagno minimo estremamente alto come 0.1, lo si dimezza con 0.05 per poi abbassarlo leggermente a 0.03. I risultati forse più interessanti si hanno quando si scende ancora con i valori 0.008, 0.0075 e 0.005. È curioso notare, dalla figura 3.1.7, come la variazione della *precision* sia maggiore passando da 0.008 a 0.0075, con un guadagno di 0.03, piuttosto che passando da 0.03 a 0.008, con un guadagno di 0.01.

	0.1000	0.0500	0.0300	0.0080	0.0075	0.0050
Accuracy	0.65	0.74	0.78	0.79	0.82	0.82
Error rate	0.35	0.26	0.22	0.21	0.18	0.18
TP	3079.00	2392.00	2984.00	2963.00	2930.00	2885.00
TPR	1.00	0.78	0.97	0.96	0.95	0.94
TNR	0.00	0.68	0.43	0.48	0.57	0.61
FPR	1.00	0.32	0.57	0.52	0.43	0.39
FNR	0.00	0.22	0.03	0.04	0.05	0.06
Precision	0.65	0.82	0.76	0.77	0.80	0.82
Recall	1.00	0.78	0.97	0.96	0.95	0.94
F1	0.79	0.80	0.85	0.86	0.87	0.87

Figura 3.1.7 Variazione delle metriche di valutazione dell’albero decisionale in base ai valori di guadagno minimo

Se si chiede all’algoritmo di operare split con un guadagno minimo di 0.1, nessuno split viene operato. Si rimane con un unico nodo dove viene semplicemente classificata la classe più frequente.

Nella figura 3.1.8 è rappresentato l'albero decisionale con min impurity decrease = 0.1

```
gini = 0.456
samples = 14265
value = [9253, 5012]
class = g
```

Figura 3.1.8 Valutazione delle metriche dell'albero con min impurity decrease=0.1 che si riduce a un solo nodo

Per il valore 0.05 viene indotto un albero molto semplice, con solo 2 livelli di profondità, 2 foglie e 1 split. Un albero così semplice è in grado di avere una purezza dello 0.82, estremamente vicina allo 0.85 dell'albero tuned, estremamente più complesso con 7 livelli di profondità, 12 foglie e 15 split. Tuttavia si ha un numero minore di TP, classificandone 2392.

Nella figura 3.1.9, albero decisionale con min impurity decrease = 0.5.

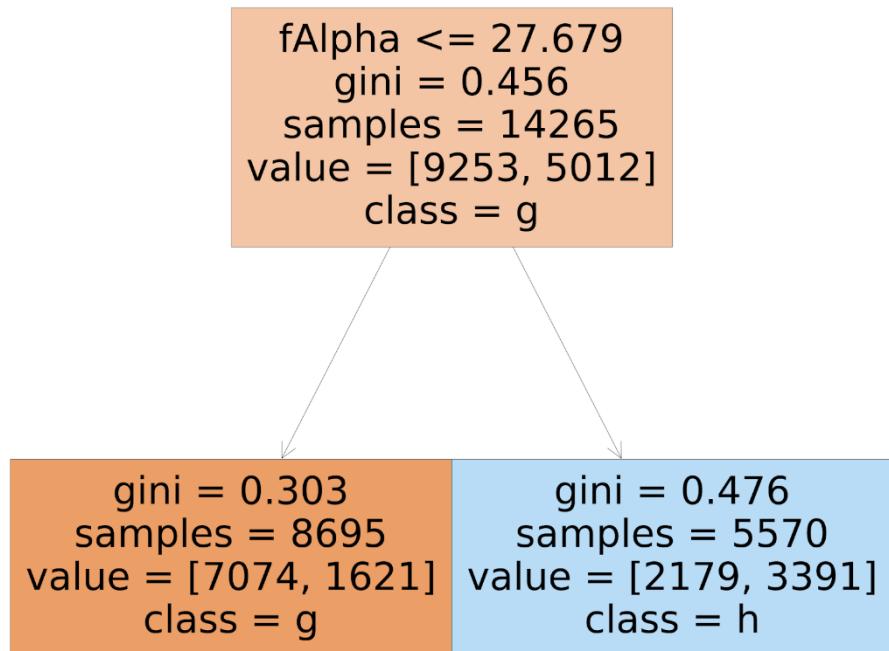


Figura 3.1.9 Albero decisionale a 2 foglie e 2 livelli di profondità generato per min impurity descrease = 0.5.

Nella figura 3.1.10, l'albero decisionale con min impurity decrease = 0.3

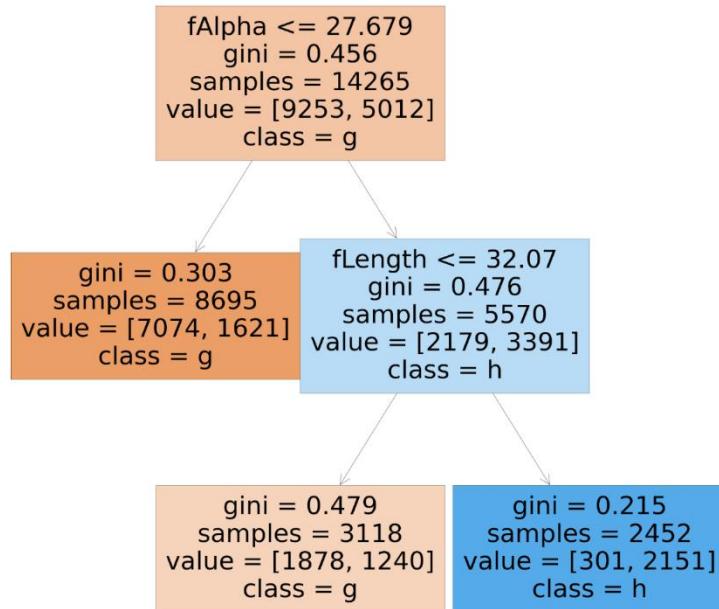


Figura 3.1.10 Albero decisionale a 3 foglie e 3 livelli di profondità generato per min impurity descrease = 0.3

Albero decisionale rappresentato in figura 3.1.11, è generato con min impurity decrease = 0.008

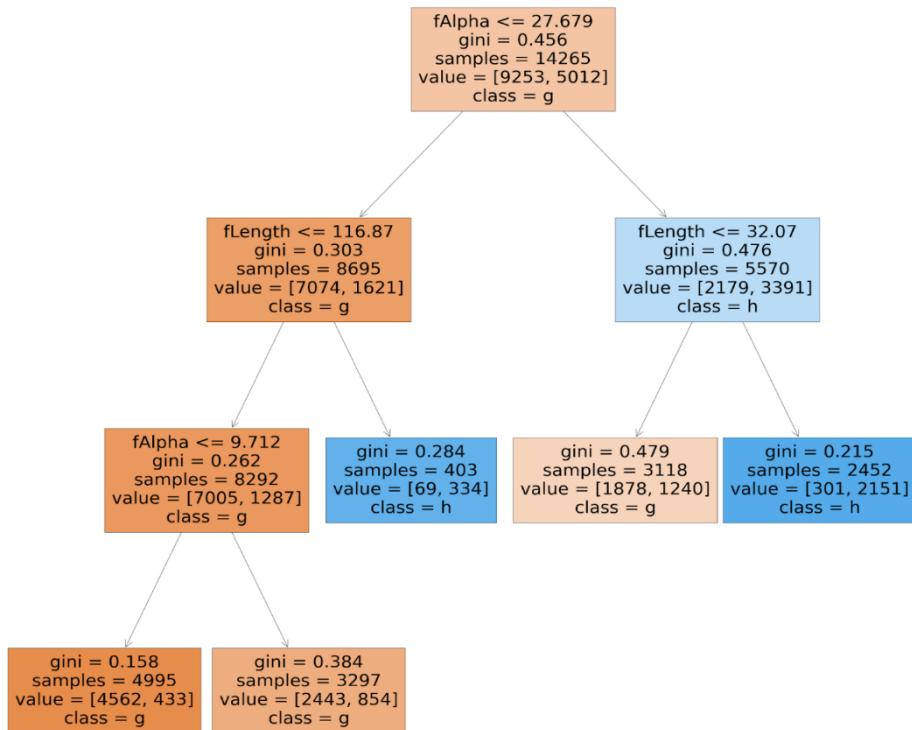


Figura 3.1.11 Albero decisionale a 5 foglie e 4 livelli di profondità generato per min impurity descrease = 0.008

Per quanto riguarda gli alberi con min impurity decrease 0.0075 e 0.005, mostrati in figura 3.1.12 e 3.1.13 rispettivamente, essi non hanno una forma estremamente semplice come i precedenti, ma neanche esageratamente complessa come il tuned. Essi hanno una complessità che ci si aspetta inducendo la classificazione su dati grezzi e utilizzando dieci attributi. Gli alberi generati con min impurity decrease 0.0075 e 0.005 hanno entrambi 5 livelli di profondità. Il primo ha con 7 nodi foglia, 6 split mentre il secondo ha 8 nodi foglia e 7 split. È interessante notare come nei grafici che plottano il numero del massimo livello di profondità dell'albero e il numero massimo di nodi foglia con la Precision questi valori sono in corrispondenza di un massimo locale, un compromesso tra la complessità del modello e buone prestazioni. Con una Precision di 0.80 e 0.82 ci si trova a livelli paragonabili all'albero tuned, e si riesce persino a classificare un numero discretamente maggiore di TP: 2930 e 2885 a confronto dei 2796 dell'albero tuned.

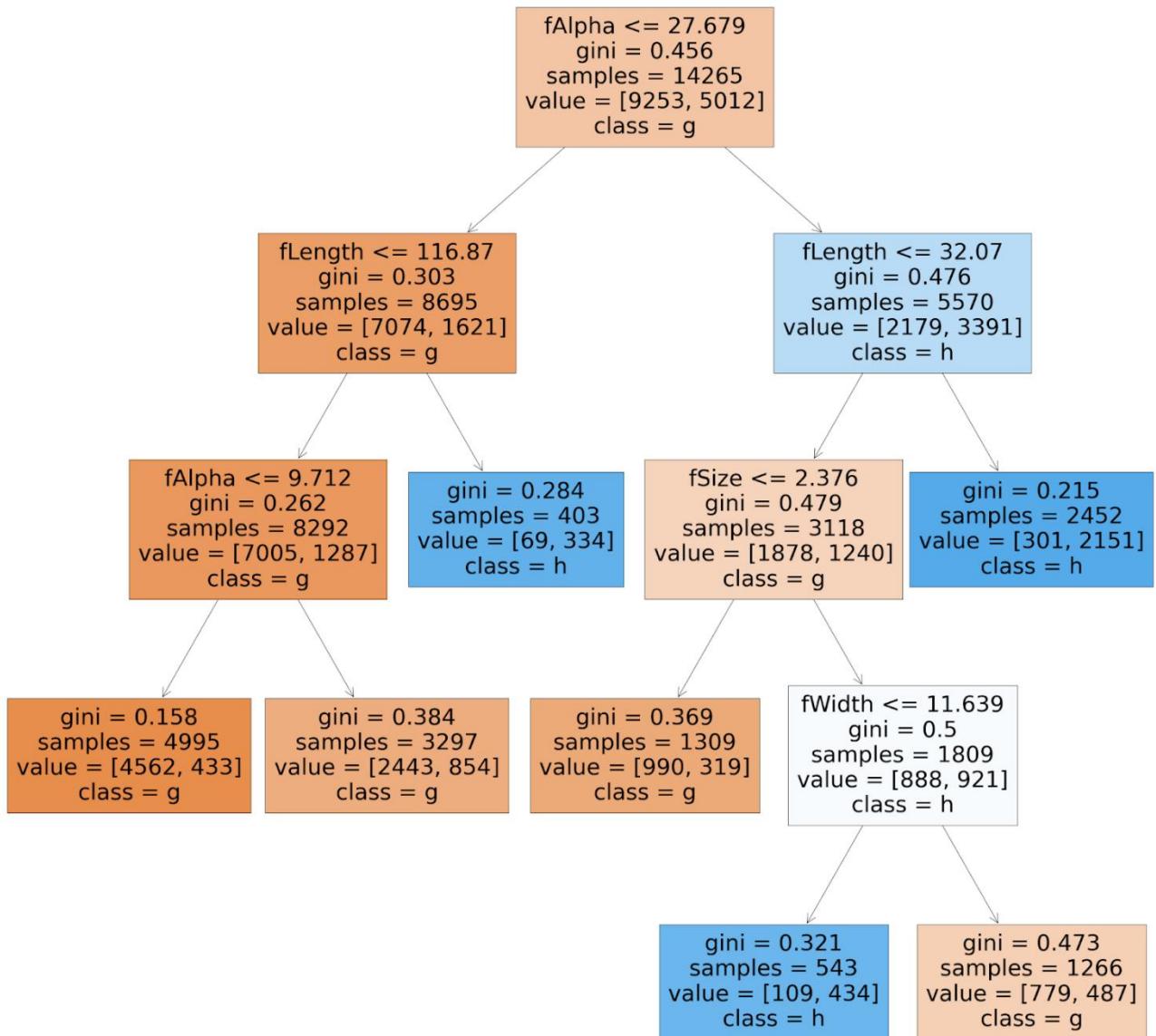


Figura 3.1.12 Albero decisionale a 7 foglie e 5 livelli di profondità generato per min impurity descrease = 0.0075

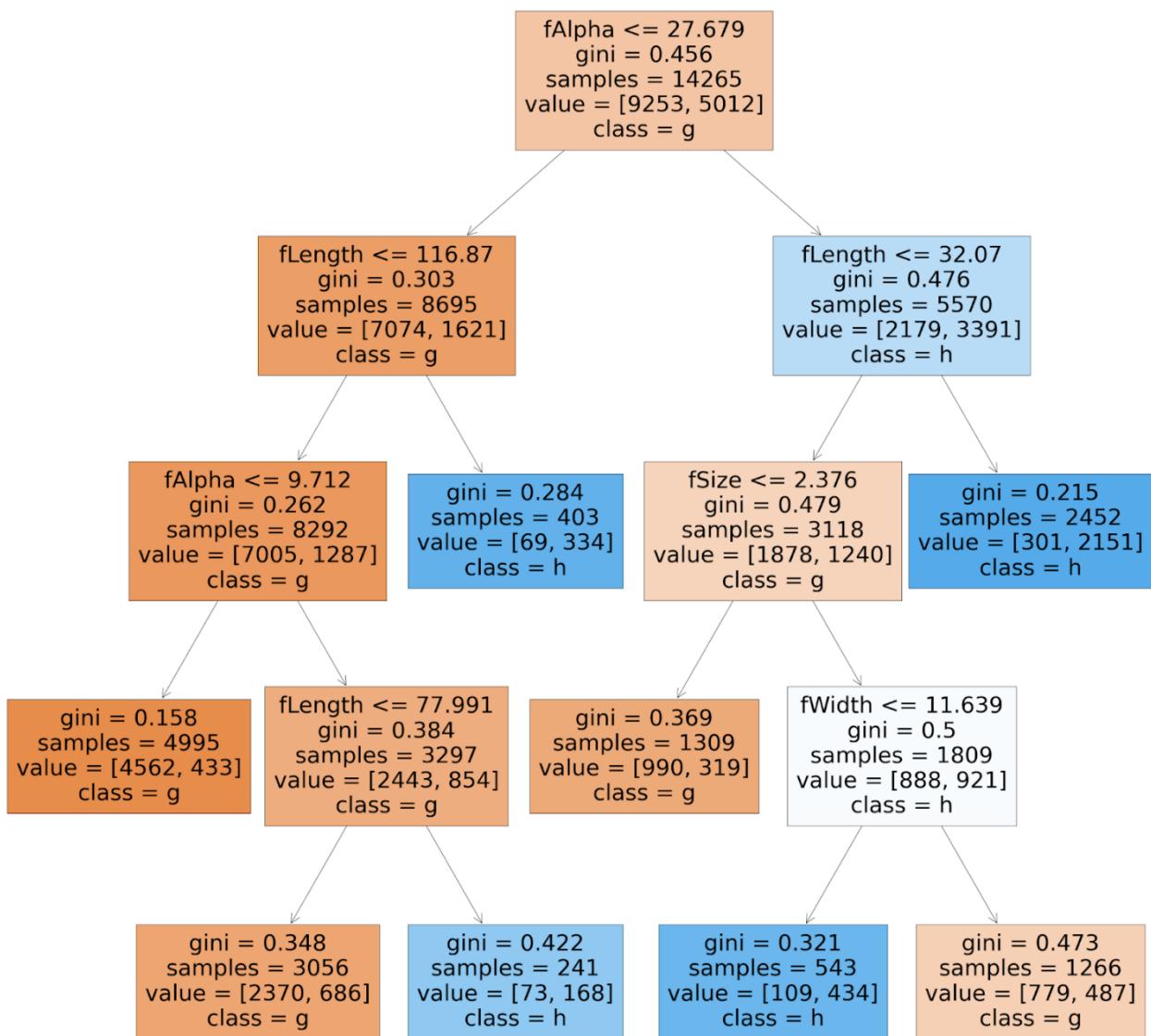


Figura 3.1.13 Albero decisionale a 8 foglie e 5 livelli di profondità generato per min impurity descrease = 0.005

3.2. Classificatore a istanze kNN

L’analisi del classificatore k-Nearest Neighbors (kNN) è stata condotta utilizzando il modello fornito da Scikit-Learn, concentrandosi sull’iperparametro chiave, ovvero il numero di vicini considerati per la classificazione di un nuovo record, indicato come k . Inizialmente, è stato esplorato un range di valori per k , da 1 a 50, allo scopo di comprendere quanto l’accuratezza e precisione del modello variassero al crescere dell’iperparametro. Tuttavia, questa analisi ha evidenziato che la stabilità delle metriche si raggiunge ben prima di $k = 50$, infatti si è successivamente optato per un range di valori da 1 a 20.

È necessario considerare che questo processo iniziale di classificazione è stato eseguito utilizzando il dataset grezzo, quindi includendo tutti i 10 attributi disponibili e senza applicare alcun tipo di bilanciamento al training set. Tutti i dati mostrati a seguire sono quindi relativi al dataset grezzo e possibilmente fuorvianti sulle vere possibilità del modello.

Il parametro k è stato valutato in tre modalità differenti: la prima totalmente default, senza pesare le distanze ed utilizzando la distanza euclidea, la seconda applicando invece un peso alle distanze e la terza utilizzando la distanza Manhattan. Nel k-NN tuned, quello reputato il migliore secondo i risultati ottenuti, vengono combinati questi tre iperparametri secondo i valori ritenuti più opportuni.

La funzione custom che esegue il tuning di default è “**kNN_classifier()**”, che accetta quattro parametri principali: **train_x**, **test_x**, **train_y**, e **test_y**, che rappresentano gli split del dataset ottenuti tramite la funzione **train_test_split** di Scikit-Learn. L’ultimo parametro, **k_values**, è opzionale e specifica il range di valori per i quali si vuole testare il classificatore, con un valore di default pari a “**range(1, 21)**”. La funzione restituisce due array contenenti le metriche relative al test set e al training set, insieme al vettore “**k_values**” che facilita la coerenza con le altre funzioni.

Successivamente, le metriche vengono convertite in DataFrame Pandas, utilizzando la funzione custom “**metrics_df**” presente nel modulo “**metrics.py**”. Questo passaggio semplifica l’analisi e la

visualizzazione dei risultati. Nella figura 3.2.1 è riportato un esempio del DataFrame relativo al test set per il range di k 1-20 è il seguente:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.78	0.76	0.80	0.79	0.80	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	
Error rate	0.22	0.24	0.20	0.21	0.20	0.20	0.20	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	
TP	2653.00	2346.00	2771.00	2609.00	2832.00	2734.00	2866.00	2789.00	2880.00	2822.00	2890.00	2850.00	2903.00	2871.00	2914.00	2875.00	2918.00	2886.00	2918.00	2894.00
TPR	0.86	0.76	0.90	0.85	0.92	0.89	0.93	0.91	0.94	0.92	0.94	0.93	0.94	0.93	0.95	0.93	0.95	0.94	0.95	0.94
TNR	0.62	0.75	0.60	0.69	0.58	0.65	0.57	0.62	0.57	0.61	0.56	0.60	0.56	0.59	0.55	0.58	0.54	0.57	0.54	0.57
FPR	0.38	0.25	0.40	0.31	0.42	0.35	0.43	0.38	0.43	0.39	0.44	0.40	0.44	0.41	0.45	0.42	0.46	0.43	0.46	0.43
FNR	0.14	0.24	0.10	0.15	0.08	0.11	0.07	0.09	0.06	0.08	0.06	0.07	0.06	0.07	0.05	0.07	0.05	0.06	0.05	0.06
Precision	0.81	0.85	0.81	0.83	0.80	0.82	0.80	0.81	0.80	0.81	0.80	0.81	0.80	0.81	0.80	0.79	0.80	0.80	0.79	0.80
Recall	0.86	0.76	0.90	0.85	0.92	0.89	0.93	0.91	0.94	0.92	0.94	0.93	0.94	0.93	0.95	0.93	0.95	0.94	0.95	0.94
F1	0.83	0.80	0.85	0.84	0.86	0.85	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86

Figura 3.2.1 Metriche relative al test set al variare di k da 1 a 20

L'analisi del DataFrame rivela che sia l'accuracy che la precision mostrano una variazione contenuta, confermando la stabilità del modello entro il range di k selezionato. Il valore migliore in assoluto si ha per $k = 2$ con precisione 0.85, è possibile osservare che valori con precisione leggermente minore è possibile classificare un numero considerevolmente maggiore di TP: scegliendo un $k = 4$ si ha una precisione di 0.83 e classificando 2609 TP, quasi trecento in più. Salendo fino a $k = 5$ si ha una precisione di 0.80 ma si riesce a classificare ben 2832 TP, quasi cinquecento in più. Valori come $k = 5$ permettono di ottenere una precisione sempre di 0.80 ma classificano 2918 TP. La scelta di quest'ultimo valore è però stata scartata per via dell'eccessiva complessità computazionale del modello. Per avere una semplicità elevata e per tenere fede alla scelta di ottenere la precisione più alta possibile, prediligendo la qualità sopra la quantità, viene scelto come migliore il valore $k = 2$. Di seguito, è stato generato un plot, figura 3.2.2, che visualizza la precisione del modello al variare di k . Questo grafico permette di esaminare le prestazioni sia sul training set che sul test set, facilitando il confronto tra le due.

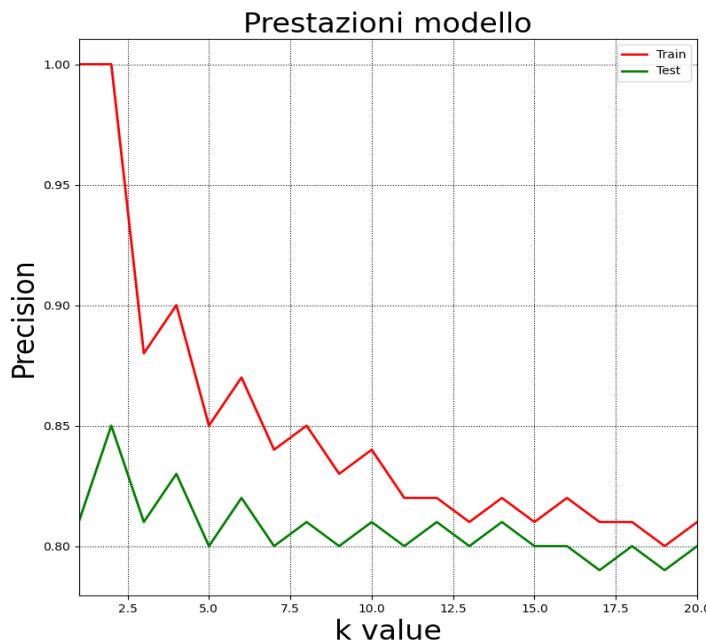


Figura 3.2.2 Precisione del modello al variare di k 1 a 20 sul test set e sul training set

Infine, è stato creato un plot, figura 3.2.3, formato da cinque subplot contenenti le curve ROC per i valori di k selezionati (1, 5, 10, 15, 20). Questo approccio consente una valutazione visiva delle performance del modello attraverso l'area sotto la curva ROC (AUC).

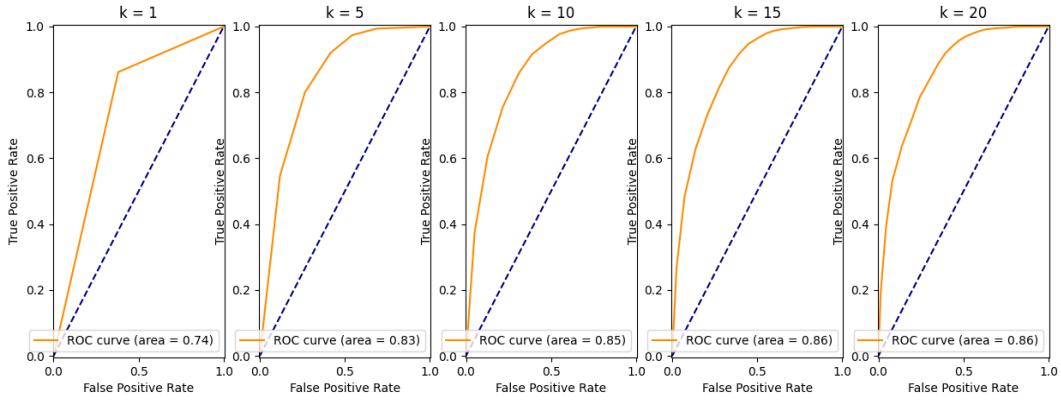


Figura 3.2.3 Curve ROC dei modelli in base al valore di k

È possibile notare che l'AUC non varia di molto da $k = 5$ in poi:

La seconda funzione custom si chiama “**kNN_weights()**”, essa esegue lo stesso tuning di k ma pesando i valori delle distanza: tra i k vicini scelti, la classe dei record più vicini a quello da classificare avrà un peso maggiore nella classificazione. Per fare ciò occorre attivare il peso con **weights= “distance”**. Qui di seguito vengono mostrate le prestazioni (figura 3.2.4), il confronto tra la precisione e il variare del k value (figura 3.2.5) e le curve ROC per i valori sopra specificati (figura 3.2.6).

Metriche di valutazione, test set:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.78	0.78	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	
Error rate	0.22	0.22	0.20	0.20	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	
TP	2653.00	2653.00	2767.00	2785.00	2830.00	2842.00	2863.00	2871.00	2884.00	2881.00	2884.00	2895.00	2898.00	2910.00	2914.00	2910.00	2917.00	2915.00	2917.00	2918.00
TPR	0.86	0.86	0.90	0.90	0.92	0.92	0.93	0.93	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
TNR	0.62	0.62	0.61	0.62	0.60	0.60	0.59	0.59	0.59	0.59	0.58	0.58	0.57	0.57	0.57	0.57	0.57	0.56	0.56	0.56
FPR	0.38	0.38	0.39	0.38	0.40	0.40	0.41	0.41	0.41	0.41	0.42	0.42	0.43	0.43	0.43	0.43	0.43	0.44	0.44	0.44
FNR	0.14	0.14	0.10	0.10	0.08	0.08	0.07	0.07	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Precision	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.80	0.81	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
Recall	0.86	0.86	0.90	0.90	0.92	0.92	0.93	0.93	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
F1	0.83	0.83	0.85	0.86	0.86	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87

il valore di k che porta precisione migliore è 1 che classifica 2653.0 TP con precisione 0.81 pesando le distanze

Figura 3.2.4

Si può notare, forse sorprendentemente, che pesare le distanze non porta a significative differenze nella precisione e neanche nell'accuratezza. Il valore reputato migliore è in questo caso il più semplice possibile: $k = 1$. Si ha un numero più basso di TP rispetto al valore senza pesi ma un'accuratezza

minore, a 0.81. E' interessante notare come la precisione sul training set sia sempre massima, questo potrebbe portare a un rischio di overfitting.

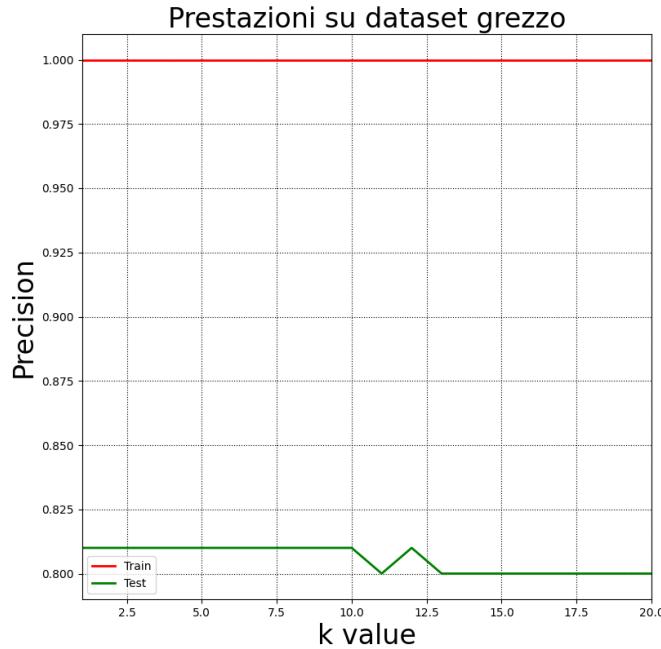


Figura 3.2.5

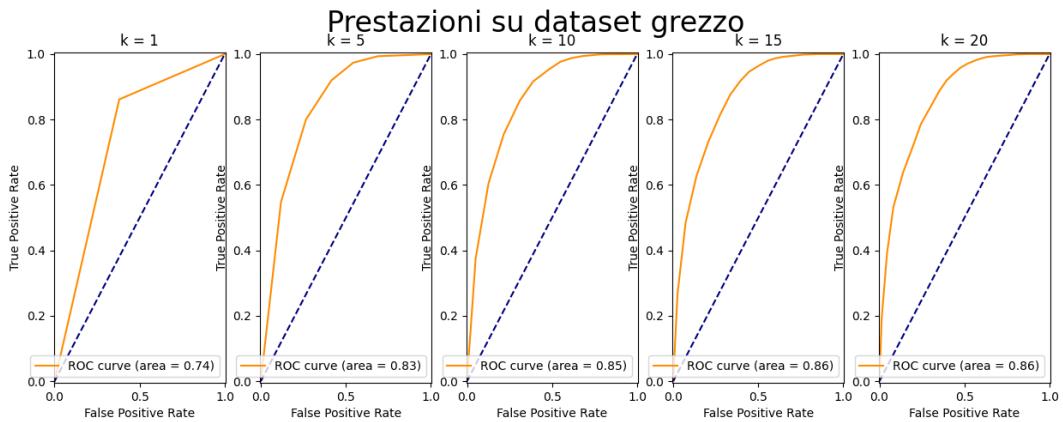


Figura 3.2.6

La terza funzione custom è “**kNN_manhattan()**”, che esegue il tuning di k ma utilizzando la distanza Manhattan invece che la distanza euclidea. Qui di seguito si mostrano la tabella con le metriche, figura 3.2.7, e il grafico k-precision, figura 3.2.8.

Metriche di valutazione, test set:																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.78	0.76	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
Error rate	0.22	0.24	0.20	0.20	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
TP	2678.00	2340.00	2803.00	2671.00	2869.00	2769.00	2897.00	2825.00	2906.00	2863.00	2914.00	2875.00	2925.00	2895.00	2938.00	2907.00	2951.00	2923.00	2945.00	2930.00
TPR	0.87	0.76	0.91	0.87	0.93	0.90	0.94	0.92	0.94	0.93	0.95	0.93	0.95	0.94	0.95	0.94	0.96	0.95	0.96	0.95
TNR	0.62	0.76	0.59	0.67	0.59	0.65	0.58	0.62	0.57	0.60	0.55	0.59	0.55	0.58	0.55	0.57	0.54	0.56	0.54	0.56
FPR	0.38	0.24	0.41	0.33	0.41	0.35	0.42	0.38	0.43	0.40	0.45	0.41	0.45	0.42	0.45	0.43	0.46	0.44	0.46	0.44
FNR	0.13	0.24	0.09	0.13	0.07	0.10	0.06	0.08	0.06	0.07	0.05	0.07	0.05	0.06	0.05	0.06	0.04	0.05	0.04	0.05
Precision	0.81	0.85	0.80	0.83	0.80	0.82	0.80	0.82	0.80	0.81	0.80	0.81	0.80	0.81	0.79	0.80	0.79	0.80	0.79	0.80
Recall	0.87	0.76	0.91	0.87	0.93	0.90	0.94	0.92	0.94	0.93	0.95	0.93	0.95	0.94	0.95	0.94	0.96	0.95	0.96	0.95
F1	0.84	0.80	0.85	0.85	0.86	0.86	0.87	0.86	0.87	0.87	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87

il valore di k che porta precisione migliore è 2 che classifica 2340.0 TP con precisione 0.85 utilizzando la distanza Manhattan

Figura 3.2.7

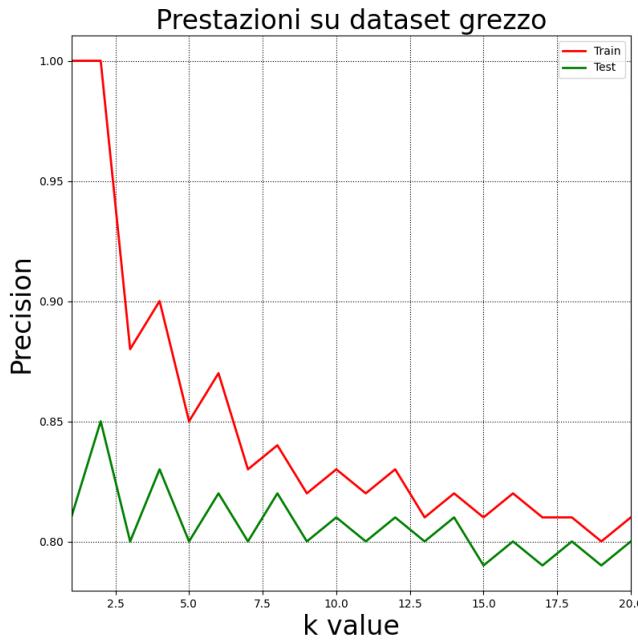


Figura 3.2.8

Le prestazioni sono così simili da essere quasi indistinguibili, il valore migliore di k è sempre 2, con una precisione di 0.85 e 2340 TP classificati. A livello di complessità computazionale la distanza Manhattan è più semplice da calcolare, non avendo elevazioni al quadrato né radici quadrate. Per questo motivo si è scelto di utilizzare questa distanza piuttosto che la euclidea.

La quarta e ultima funzione custom è “**kNN_tuned()**”, dove si induce un albero con la migliore combinazione di iperparametri decisa: si tratta di $k = 2$, **weights="uniform"** e **p = 1**. Si decide di utilizzare i due nearest neighbors, non pesare la distanza ed utilizzare la distanza Manhattan. Come notato prima, questo è il classificatore kNN che è in grado di offrire le prestazioni migliori, vedi figura 3.2.9 e 3.2.10.

```

Metriche di valutazione, test set:
                                2
Accuracy      0.76
Error rate    0.24
TP            2340.00
TPR           0.76
TNR           0.76
FPR           0.24
FNR           0.24
Precision     0.85
Recall        0.76
F1            0.80
il valore di k che porta precisione migliore è 2 che classifica 2340.0 TP con precisione 0.85 con k=2, weight=uniform e p=1

```

Figura 3.2.9 Metriche della funzione kNN_tuned()

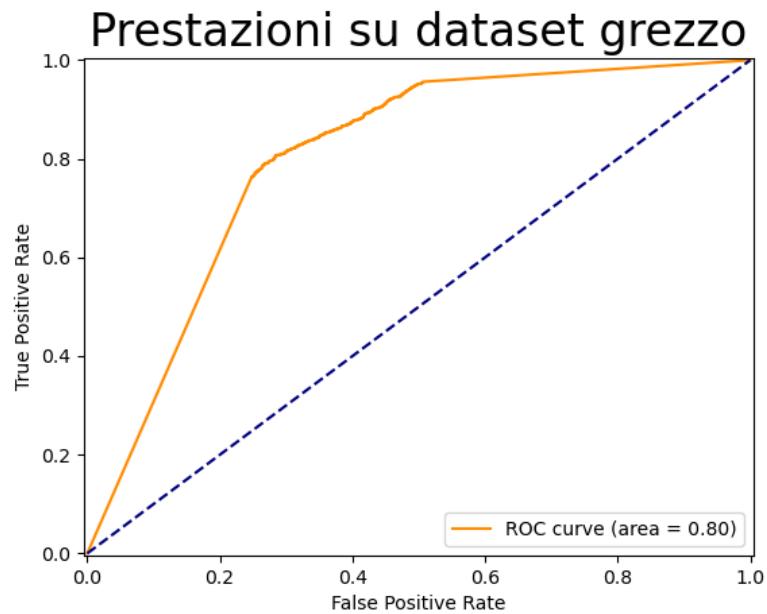


Figura 3.2.10 Prestazioni della funzione kNN_tuned()

3.3. Classificatore custom: classificatore multiplo

Come modello di classificazione custom è stato deciso di realizzare un classificatore multiplo. È stato creato un oggetto chiamato “**multiple_clf**”, che contiene al suo interno le funzioni “`__init__()`”, “`fit()`”, “`single_predict()`” e “`hard_voting()`”.

In “`__init__()`” è possibile passare in una lista i classificatori scelti per il classificatore multiplo. Questa funzionalità del modello viene eseguita solo quando viene chiamata direttamente la classe, infatti è anche chiamato metodo costruttore. In questa fase vengono inseriti tre classificatori in una lista. È stato deciso di utilizzare degli alberi decisionali e i modelli scelti sono i rappresentanti di tre possibili scuole di pensiero. Il primo scelto è l’albero tuned, ottenuto attraverso il tuning degli iperparametri `max_depth` e `min_leaf_nodes` e realizzato con i valori che massimizzano la precisione, portandola a 0.85. Esso rappresenta il tentativo di scegliere i valori migliori possibili senza curarsi troppo della complessità del modello indotto. Il secondo scelto è l’albero che è stato ritenuto il migliore: si tratta del `min gain 0.005`. Una via di mezzo tra le prestazioni e la complessità computazionale, che è in grado di avere alta precisione di 0.82 con un modello più contenuto. Il terzo scelto è invece l’altro estremo, il `min gain 0.05`: una precisione alta a 0.82 con il modello di albero più semplice possibile, semplicemente uno split con due foglie. Questo albero privilegia la semplicità del modello rispetto alle prestazioni, poiché nonostante la precisione sia alta il numero di TP classificato potrebbe essere migliorato.

Si crede che l’unione di questi tre approcci possa portare a un modello in grado di avere una complessità contenuta, mediando i modelli, con una precisione più alta dei singoli classificatori.

In “`fit()`” vengono indotti i tre alberi e in “`single_predict()`” essi predicono singolarmente le classi. Nella funzione “`hard_voting()`” viene infine effettuata l’effettiva previsione, scegliendo la classe più frequente tra quelle predette in “`single_predict()`”.

Qui di seguito le tabelle, figura 3.3.1, con le prestazioni dei singoli classificatori e del classificatore multiplo. `Clf_1` è l’albero tuned, `Clf_2` è l’albero `min gain 0.005` e `Clf_3` è l’albero `min gain 0.05`.

Eseguo classificatore multiplo custom su dataset grezzo...				
Metriche di valutazione, test set, su dataset grezzo:				
	Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.83	0.82	0.74	0.83
Error rate	0.17	0.18	0.26	0.17
TP	2796.00	2885.00	2392.00	2831.00
TPR	0.91	0.94	0.78	0.92
TNR	0.70	0.61	0.68	0.66
FPR	0.30	0.39	0.32	0.34
FNR	0.09	0.06	0.22	0.08
Precision	0.85	0.82	0.82	0.83
Recall	0.91	0.94	0.78	0.92
F1	0.88	0.87	0.80	0.87

Figura 3.3.1 Prestazioni dei singoli classificatori e del classificatore combinato

La precisione è maggiore rispetto a quella dei singoli modelli meno complessi, ma non arriva ai livelli dell'albero tuned. Il numero dei TP invece aumenta di parecchio, fino ad arrivare quasi ai livelli dell'albero che classifica più TP. Complessivamente, almeno sui dati grezzi, i risultati ottenuti non giustificano l'utilizzo del classificatore multiplo: il classificatore min gain 0.005 è sempre da reputare il migliore. Si osserverà se con l'utilizzo dei dati pre processati i risultati potranno cambiare.

4. Pre-processing

Data pre-processing:

Per pre processare i dati si è scelto di attuare quattro tecniche separate, per poter vedere come cambiano le prestazioni dei classificatori dopo ognuna di esse. In seguito si valuterà la migliore accoppiata di tecniche con il miglior classificatore scelto. Queste tecniche sono: bilanciamento (attraverso undersampling e oversampling), standardizzazione, selezione features e aggregazione di features.

Aggregazione di features:

Per effettuare il data pre-processing è necessario, nel nostro caso, ridurre il numero di attributi utilizzati. Algoritmi come il k-NN funzionano meglio con meno attributi, poiché la differenza tra distanza massima e minima diminuisce con l'aumentare delle dimensioni. Uno dei modi per fare ciò è sfruttare la PCA, ovvero la *Principal Component Analysis*. Grazie ad un algoritmo interno a Scikitlearn fa una selezione degli attributi migliori da usare in base ai loro valori di varianza, combinando i loro valori per evitare delle correlazioni e far sì che in poche variabili siano comprese le informazioni più importanti. Quando si seleziona come pre-processing l'aggregazione feature, viene richiesta all'utente la quantità di attributi che si desidera ottenere. Tutte le immagini relative a questo tipo di pre-processing sono state ottenute con un'aggregazione attributi tale da ottenerne 3.

Selezione di features

Per una feature selection è stato utile controllare la presenza di valori unici per ogni attributo, e per fSize, fConc e fConc1 i valori sono notevolmente più bassi rispetto agli altri. Per il modello con l'albero decisionale eliminare tre attributi può ritenersi sufficiente, nonostante la scelta degli attributi sia di tipo *embedded* e pertanto interna all'algoritmo, una riduzione preventiva può facilitare la qualità degli split. Per il k-NN conviene diminuire ulteriormente il numero degli attributi per poterne evitare la dispersione. La scelta dei tre attributi da eliminare è una conseguenza della bassa varianza di questi ultimi, che possono perciò essere esclusi a priori. Una delle feature che invece risulta migliore, in base agli istogrammi, è fAlpha, che per i suoi valori separa in maniera netta la classe *g* dalla *h*.

Osservando la matrice di correlazione risulta che i tre attributi nominati prima hanno valori molto alti, perciò una correlazione elevata che li rende molto simili tra di loro, mentre degli attributi che

hanno dei valori vicini allo zero indicherebbero una bassa correlazione, e perciò una grande utilità per l'algoritmo. Tra gli attributi che hanno molti valori bassi troviamo nuovamente fAlpha, ma anche fM3Trans e, in minor parte, fM3Long. Questi tre attributi potrebbero essere selezionati per la creazione del k-NN.

Bilanciamento

Essendo il dataset sbilanciato occorre anche effettuare un'azione di oversampling o undersampling, e per completezza sono state eseguite entrambe con 3 algoritmi ciascuna. Nella presentazione del dataset è segnalato il fatto che la classe h sia sottostimata, per valorizzare la classe positiva g. I dati relativi a quest'ultima sono perciò più preziosi e non conviene ridurli, mentre i dati relativi alla classe h possono essere creati sinteticamente per bilanciare il dataset.

Per l'oversampling converrebbe usare SMOTE oppure ADASYN come funzioni di Scikitlearn, le quali sfruttano dei metodi non casuali più precisi; soprattutto l'ultimo riesce a creare una decision boundary più robusta. Per l'undersampling sono state implementate e utilizzate le seguenti tecniche: random, probabilistico, e NearMiss_v1. Per quanto riguarda la tecnica probabilistica abbiamo notato che solitamente aumenta le prestazioni del modello utilizzato, soprattutto per il classificatore multiplo custom. Tra tutte le tecniche proposte, la NearMiss_v1 risulta essere quella della quale è più semplice notare il cambiamento visivo del grafico.

Standardizzazione

I dieci attributi presentano notevoli differenze anche per quanto riguarda gli ordini di grandezza dei dati che li caratterizzano. Si hanno valori inferiori all'uno come fConc ed fConc1, nell'ordine delle unità come fSize, nell'ordine delle centinaia come fDist, mentre gli altri attributi si stabilizzano nell'ordine delle decine. Per il funzionamento di un algoritmo come il k-NN, totalmente basato su distanze, è fondamentale che gli attributi rientrino nello stesso ordine di grandezza. Per questo motivo si è scelto di utilizzare la standardizzazione Min Max, dove a ogni dato viene sottratto il suo massimo, dividendo. In questo modo ogni valore è compreso in un range tra 0 e 1.

Nota all'analisi delle prestazioni su dati pre-processati:

Per tutti i classificatori sono stati eseguiti diversi tuning degli iperparametri, come mostrato nel capitolo precedente, per poi arrivare alla versione ritenuta migliore, chiamata tuned. Le stesse funzioni sono state applicate a qualsiasi coppia dati pre processati - classificatore, ottenendo per ogni coppia: grafici sull'andamento degli iperparametri a paragone con la precisione, curve ROC, e tabelle con tutte le metriche a confronto. Vista la mole di dati è stato scelto di non metterli tutti nella relazione, ma di presentare i più significativi con delle note di commento. Per la consultazione di tali dati è possibile accedere alla cartella di immagini, dove sono state divise per argomento. La cartella precisa è images/analisi dei singoli clf.

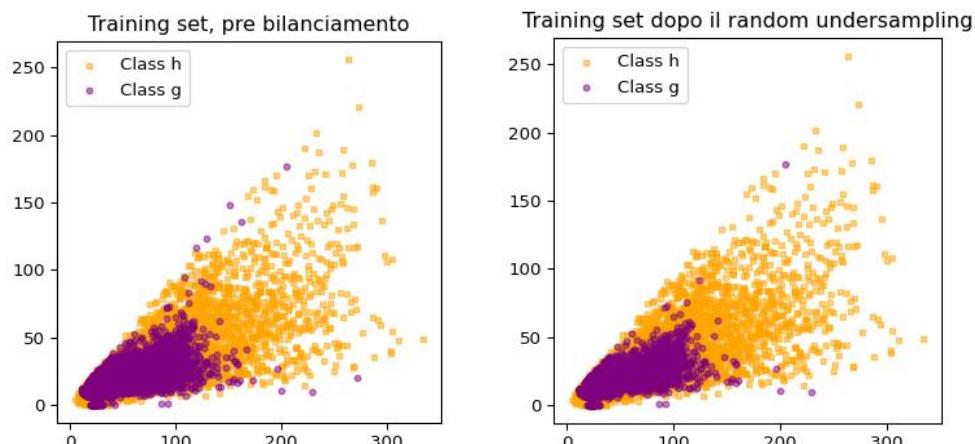
4.1. Confronto Bilanciamento

Classificatore a instanza: kNN

Undersampling

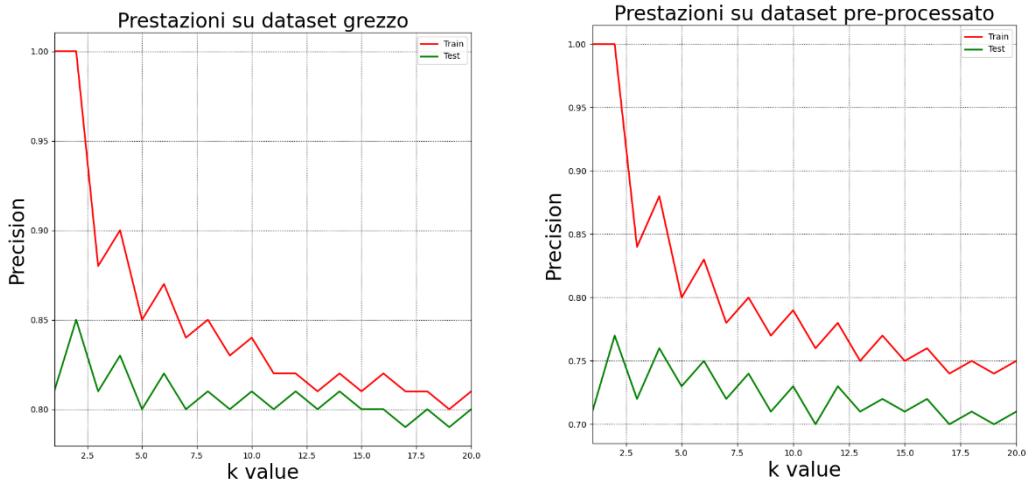
Random Undersampler:

Nell'undersampling sono state utilizzate tre diverse tecniche per diminuire il numero di record della classe maggioritaria. Nel primo caso è stato usato **RandomUnderSampler**, che elimina casualmente gli elementi della classe maggioritaria. Nel visualizzare la distribuzione dei record prima e dopo il bilanciamento non si nota un cambiamento eccessivo, proprio per via del fatto che i dati sono eliminati randomicamente.



In questo caso l'andamento del classificatore è simile, però sui dati grezzi le prestazioni risultano migliori. Questo perché è stata effettuata solo una diminuzione del numero di record senza agire sugli attributi, cambiando il rapporto tra elementi della classe positiva e negativa.

Prestazioni kNN con diversi k value



Anche nelle altre metriche di valutazione si può notare un calo nelle prestazioni rispetto a quelle ottenute con i dati grezzi, sempre a causa del numero ridotto di record usati dal modello per l'addestramento.

Qui di seguito il confronto per effettuare il tuning degli iperparametri modificando: iperparametro k (prima immagine), pesando le distanze (seconda immagine), e usando la distanza Manhattan invece di quella Euclidea (terza immagine). Infine si è selezionato il miglior valore (quarta immagine) con k=2, uniformità di pesi delle distanze e distanza Manhattan.

Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.73	0.72	0.75	0.76	0.76	0.76	0.76	0.76	0.75	0.76	0.75	0.76	0.76	0.76	0.75	0.75	0.75	0.75	0.75	0.75	
Error rate	0.27	0.28	0.25	0.24	0.24	0.24	0.24	0.24	0.25	0.24	0.25	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.25	0.25	
TP	980.00	803.00	1037.00	951.00	1065.00	988.00	1074.00	1022.00	1076.00	1040.00	1082.00	1052.00	1094.00	1060.00	1089.00	1060.00	1086.00	1063.00	1095.00	1066.00	
TPR	0.78	0.64	0.82	0.75	0.84	0.78	0.85	0.81	0.85	0.76	0.82	0.86	0.83	0.87	0.84	0.86	0.84	0.86	0.84	0.87	0.85
TNR	0.68	0.80	0.68	0.76	0.68	0.73	0.66	0.71	0.65	0.69	0.63	0.69	0.64	0.68	0.64	0.67	0.63	0.66	0.62	0.66	0.66
FPR	0.32	0.20	0.32	0.24	0.32	0.27	0.34	0.29	0.35	0.31	0.37	0.31	0.36	0.32	0.36	0.33	0.37	0.34	0.38	0.34	0.34
FNR	0.22	0.36	0.18	0.25	0.16	0.22	0.15	0.19	0.15	0.18	0.14	0.17	0.13	0.16	0.14	0.16	0.14	0.16	0.13	0.15	0.15
Precision	0.71	0.77	0.72	0.76	0.73	0.75	0.72	0.74	0.71	0.73	0.70	0.73	0.71	0.72	0.71	0.72	0.70	0.71	0.70	0.71	0.71
Recall	0.78	0.64	0.82	0.75	0.84	0.78	0.85	0.81	0.85	0.82	0.86	0.83	0.87	0.84	0.86	0.84	0.86	0.84	0.87	0.85	0.85
F1	0.74	0.70	0.77	0.76	0.78	0.76	0.78	0.77	0.77	0.78	0.77	0.78	0.78	0.78	0.78	0.78	0.77	0.77	0.78	0.77	0.77
il valore di k che porta precisione migliore è 2 che classifica 803.0 TP con precisione 0.77 senza cambiare alcun iperparametro eccetto k																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.73	0.73	0.75	0.76	0.77	0.76	0.77	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.75	0.75	0.76	0.76
Error rate	0.27	0.27	0.25	0.24	0.23	0.24	0.24	0.23	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.25	0.24	0.25	0.25	0.24
TP	980.00	980.00	1037.00	1047.00	1070.00	1068.00	1081.00	1088.00	1080.00	1086.00	1084.00	1091.00	1093.00	1094.00	1092.00	1090.00	1092.00	1093.00	1097.00	1103.00	
TPR	0.78	0.78	0.82	0.83	0.85	0.85	0.86	0.86	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.86	0.87	0.87	0.87	0.87	0.87
TNR	0.68	0.68	0.68	0.69	0.69	0.67	0.67	0.67	0.66	0.65	0.66	0.65	0.65	0.65	0.65	0.65	0.65	0.64	0.64	0.64	0.65
FPR	0.32	0.32	0.32	0.31	0.31	0.33	0.33	0.33	0.34	0.34	0.35	0.34	0.35	0.34	0.35	0.34	0.35	0.35	0.36	0.36	0.35
FNR	0.22	0.22	0.18	0.17	0.15	0.15	0.14	0.14	0.14	0.14	0.13	0.13	0.13	0.13	0.13	0.14	0.13	0.13	0.13	0.13	0.13
Precision	0.71	0.71	0.72	0.73	0.74	0.73	0.72	0.73	0.72	0.72	0.71	0.72	0.72	0.72	0.71	0.72	0.71	0.71	0.71	0.71	0.72
Recall	0.78	0.78	0.82	0.83	0.85	0.85	0.86	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
F1	0.74	0.74	0.77	0.78	0.79	0.78	0.78	0.79	0.78	0.78	0.79	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.79	0.79
il valore di k che porta precisione migliore è 5 che classifica 1070.0 TP con precisione 0.74 pesando le distanze																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.73	0.73	0.76	0.77	0.75	0.77	0.76	0.77	0.76	0.76	0.77	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.77	0.76	0.76
Error rate	0.27	0.27	0.24	0.23	0.25	0.23	0.24	0.23	0.24	0.24	0.23	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.23	0.24
TP	1004.00	827.00	1073.00	979.00	1079.00	1024.00	1092.00	1043.00	1094.00	1053.00	1108.00	1076.00	1109.00	1072.00	1107.00	1081.00	1107.00	1085.00	1113.00	1087.00	
TPR	0.80	0.66	0.85	0.78	0.86	0.81	0.87	0.83	0.87	0.84	0.88	0.85	0.88	0.85	0.88	0.86	0.88	0.86	0.88	0.86	0.86
TNR	0.67	0.81	0.67	0.76	0.65	0.73	0.66	0.71	0.65	0.69	0.64	0.68	0.64	0.68	0.64	0.67	0.64	0.67	0.65	0.66	0.66
FPR	0.33	0.19	0.33	0.24	0.35	0.27	0.34	0.29	0.35	0.31	0.36	0.32	0.36	0.32	0.36	0.33	0.36	0.33	0.35	0.34	0.34
FNR	0.20	0.34	0.15	0.22	0.14	0.19	0.13	0.17	0.13	0.16	0.12	0.15	0.12	0.15	0.12	0.14	0.12	0.14	0.12	0.14	0.14
Precision	0.71	0.78	0.72	0.77	0.71	0.75	0.72	0.74	0.72	0.73	0.71	0.73	0.71	0.73	0.71	0.72	0.71	0.72	0.72	0.72	0.72
Recall	0.80	0.66	0.85	0.78	0.86	0.81	0.87	0.83	0.87	0.84	0.88	0.85	0.88	0.85	0.88	0.86	0.88	0.86	0.88	0.86	0.86
F1	0.75	0.71	0.78	0.77	0.78	0.78	0.79	0.78	0.78	0.78	0.79	0.79	0.78	0.79	0.78	0.79	0.79	0.79	0.79	0.79	0.79
il valore di k che porta precisione migliore è 2 che classifica 827.0 TP con precisione 0.78 utilizzando la distanza Manhattan																					

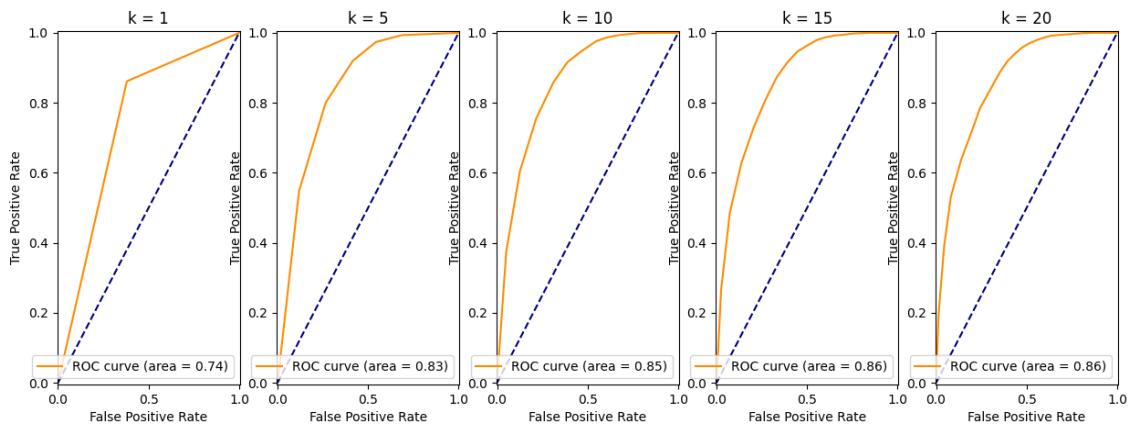
```

Metriche di valutazione, test set:
2
Accuracy      0.73
Error rate    0.27
TP            827.00
TPR           0.66
TNR           0.81
FPR           0.19
FNR           0.34
Precision     0.78
Recall        0.66
F1            0.71
il valore di k che porta precisione migliore è 2 che classifica 827.0 TP con precisione 0.78 con k=2, weight=uniform e p=1

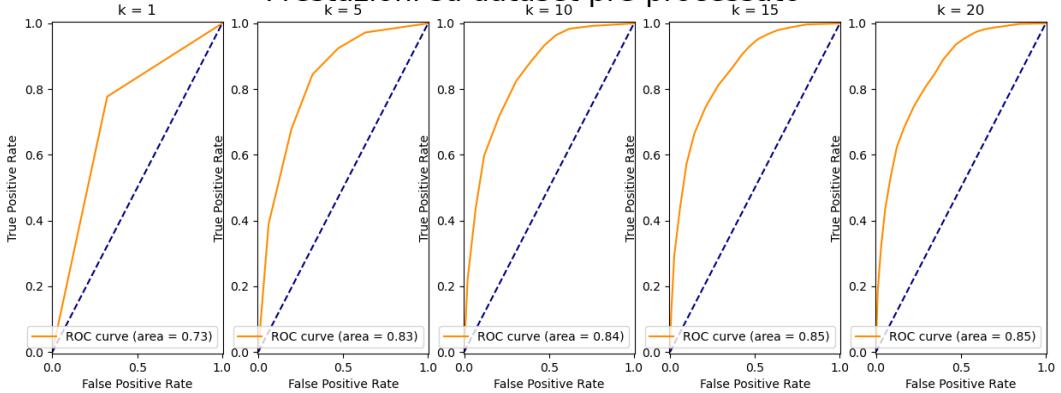
```

Per il calcolo della curva ROC è stato tracciato il plot sia con i dati non processati (immagine in alto) che con quelli pre-processati (immagine in basso). Si nota che le aree sotto alle curve della prima immagine risultano essere quasi uguali alle aree sotto alle curve ottenute dai dati grezzi; il valore dell'AUC diminuisce solo di 0.01 per quasi tutti i valori. Per valutare nella maniera più completa possibile si è scelto di implementare, per i dati pre-processati, oltre alla distanza euclidea di default, anche una distanza euclidea che effettui una media pesata delle distanze in base alla vicinanza con l'oggetto da classificare; un altro tipo di distanza che si è scelto di usare è la distanza Manhattan. Il grafico della curva ROC risulta perfettamente identico per tutti i tre metodi scelti .

Prestazioni su dataset grezzo

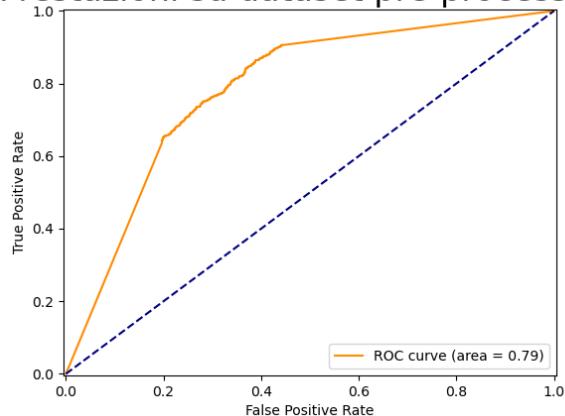


Prestazioni su dataset pre-processato



Prestazioni kNN tuned

Prestazioni su dataset pre-processato

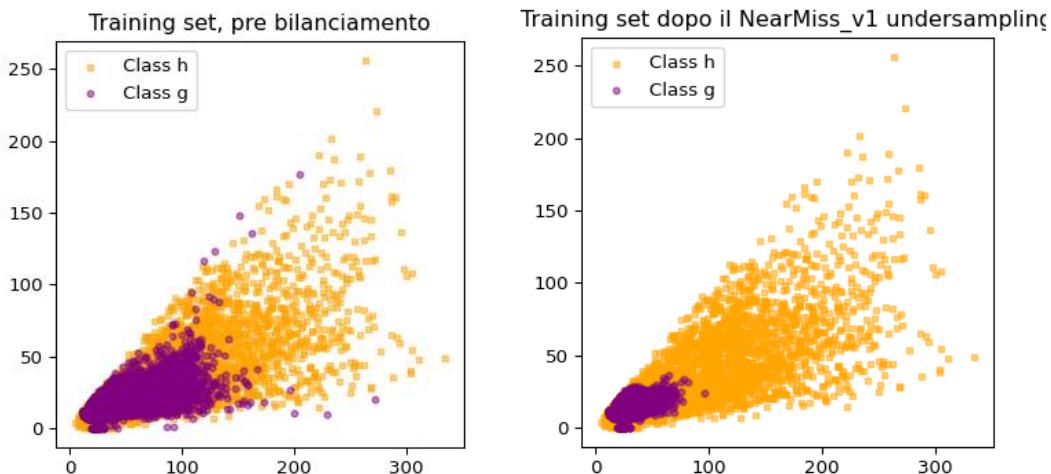


Infine sui dati finali processati e sui quali è stato trovato il miglior bilanciamento con il RandomUndersampler l'andamento della curva ROC è il seguente. Il valore di k è pari a due e non sono state pesate le distanze, utilizzando la distanza Manhattan; la precision di questo modello vale 0.78. Risulta perciò il miglior compromesso tra complessità e prestazioni del modello.

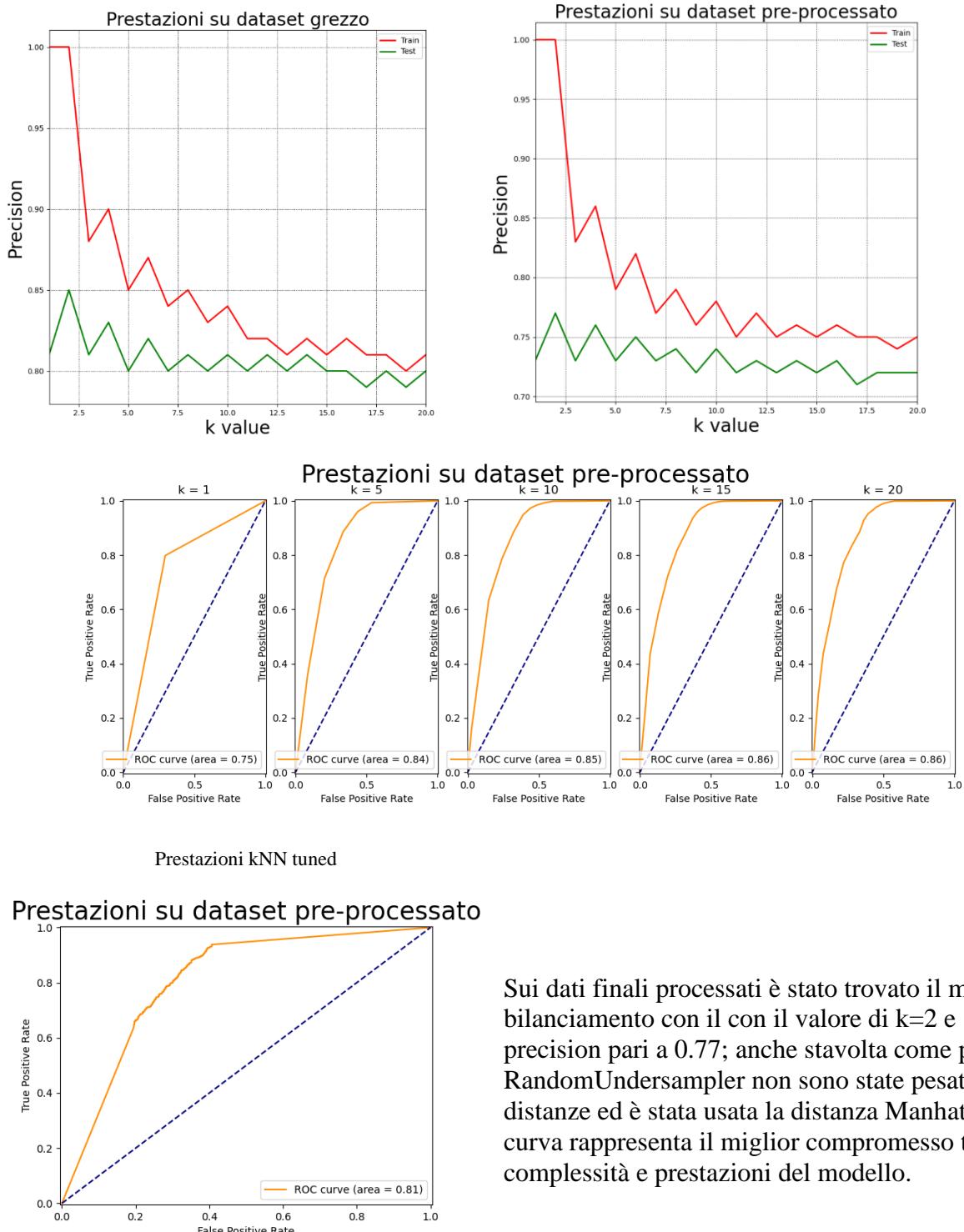
Nearmiss:

La seconda strategia usata è quella con **NearMiss**, la quale seleziona gli elementi della classe maggioritaria più vicini a quelli della classe minoritaria per eliminarli. In tale modo la classe minoritaria risulta avere più “spazio” e i suoi record possono essere classificati con maggior chiarezza.

Di seguito la distribuzione dei record prima e dopo l'undersampling, e come si vede mostrato chiaramente nelle immagini i record di classe g diminuiscono notevolmente e risultano molto più concentrati dei record h , favorendo perciò l'aumento della precision menzionato prima.



Nearmiss aumenta in maniera abbastanza significativa il rendimento delle metriche (nelle immagini è mostrato l'andamento della precisione), mentre ha delle prestazioni quasi uguali al modello che usa i dati grezzi per quanto riguarda la curva ROC. Anche in questo caso sono state tracciate più curve usando distanza euclidea con e senza pesi, e la distanza Manhattan; i risultati sono tutti molto simili tra di loro come nel RandomUndersampler.



Sui dati finali processati è stato trovato il miglior bilanciamento con il valore di $k=2$ e precision pari a 0.77; anche stavolta come per il RandomUndersampler non sono state pesate le distanze ed è stata usata la distanza Manhattan. La curva rappresenta il miglior compromesso tra complessità e prestazioni del modello.

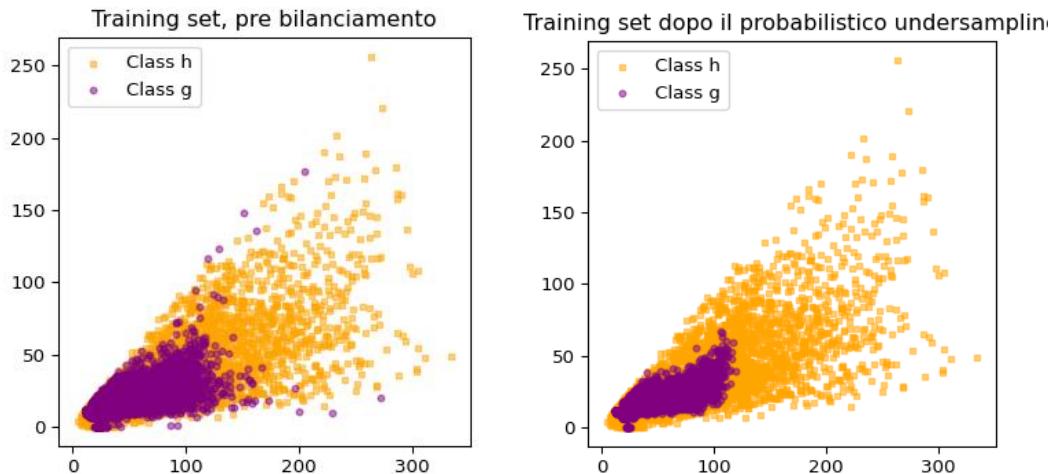
Confronto per effettuare il tuning degli iperparametri modificando: iperparametro k (prima immagine), pesando le distanze (seconda immagine), e usando la distanza Manhattan invece di quella Euclidea (terza immagine). Infine si è selezionato il miglior valore (quarta immagine).

Metriche di valutazione, test set:																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
Accuracy	0.75	0.72	0.77	0.77	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78		
Error rate	0.25	0.28	0.23	0.23	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22		
TP	1007.00	805.00	1089.00	994.00	1118.00	1064.00	1137.00	1094.00	1147.00	1115.00	1165.00	1135.00	1170.00	1144.00	1181.00	1154.00	1185.00	1166.00	1189.00	1173.00					
TPR	0.80	0.64	0.86	0.79	0.89	0.84	0.90	0.87	0.91	0.88	0.92	0.90	0.93	0.91	0.94	0.92	0.94	0.92	0.94	0.94	0.93				
TNR	0.70	0.80	0.68	0.74	0.66	0.71	0.66	0.69	0.65	0.68	0.64	0.67	0.63	0.65	0.62	0.64	0.62	0.64	0.62	0.64	0.62	0.64			
FPR	0.30	0.20	0.32	0.26	0.34	0.29	0.34	0.31	0.35	0.32	0.36	0.33	0.37	0.35	0.37	0.35	0.38	0.36	0.36	0.38	0.36	0.36			
FNR	0.20	0.36	0.14	0.21	0.11	0.16	0.10	0.13	0.09	0.12	0.08	0.10	0.07	0.09	0.06	0.08	0.06	0.08	0.06	0.07	0.06	0.07			
Precision	0.73	0.77	0.73	0.76	0.73	0.75	0.73	0.74	0.72	0.74	0.72	0.73	0.72	0.73	0.72	0.73	0.71	0.72	0.72	0.72	0.72	0.72			
Recall	0.80	0.64	0.86	0.79	0.89	0.84	0.90	0.87	0.91	0.88	0.92	0.90	0.93	0.91	0.94	0.92	0.94	0.92	0.94	0.93					
F1	0.76	0.70	0.79	0.77	0.80	0.79	0.80	0.80	0.81	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81			
il valore di k che porta precisione migliore è 2 che classifica 805.0 TP con precisione 0.77 senza cambiare alcun iperparametro eccetto k																									
Metriche di valutazione, test set:																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
Accuracy	0.75	0.75	0.77	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79		
Error rate	0.25	0.25	0.23	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21		
TP	1007.00	1007.00	1084.00	1083.00	1116.00	1123.00	1135.00	1143.00	1148.00	1160.00	1159.00	1167.00	1170.00	1180.00	1181.00	1187.00	1183.00	1188.00	1186.00						
TPR	0.80	0.80	0.86	0.86	0.89	0.89	0.90	0.90	0.91	0.91	0.92	0.92	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94			
TNR	0.70	0.70	0.68	0.69	0.67	0.67	0.66	0.67	0.66	0.66	0.65	0.65	0.65	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64			
FPR	0.30	0.30	0.32	0.31	0.33	0.33	0.34	0.33	0.34	0.34	0.35	0.35	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36			
FNR	0.20	0.20	0.14	0.14	0.11	0.11	0.10	0.10	0.09	0.09	0.08	0.08	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06			
Precision	0.73	0.73	0.73	0.74	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73			
Recall	0.80	0.80	0.86	0.86	0.89	0.89	0.90	0.90	0.91	0.91	0.92	0.92	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94			
F1	0.76	0.76	0.79	0.79	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81			
il valore di k che porta precisione migliore è 4 che classifica 1083.0 TP con precisione 0.74 pesando le distanze																									
Metriche di valutazione, test set:																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20					
Accuracy	0.76	0.73	0.77	0.78	0.78	0.78	0.78	0.78	0.79	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79		
Error rate	0.24	0.27	0.23	0.22	0.22	0.22	0.22	0.22	0.21	0.22	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21		
TP	1031.00	833.00	1099.00	1016.00	1143.00	1075.00	1145.00	1108.00	1171.00	1131.00	1182.00	1154.00	1191.00	1167.00	1197.00	1175.00	1199.00	1184.00	1198.00	1187.00					
TPR	0.82	0.66	0.87	0.81	0.91	0.85	0.91	0.88	0.93	0.90	0.94	0.92	0.94	0.93	0.95	0.93	0.95	0.93	0.95	0.94	0.95	0.94			
TNR	0.70	0.80	0.67	0.74	0.65	0.70	0.64	0.69	0.64	0.67	0.63	0.67	0.64	0.66	0.63	0.65	0.63	0.65	0.65	0.63	0.65	0.63			
FPR	0.30	0.20	0.33	0.26	0.35	0.30	0.36	0.31	0.36	0.33	0.37	0.33	0.36	0.34	0.37	0.35	0.37	0.35	0.37	0.35	0.37	0.35			
FNR	0.18	0.34	0.13	0.19	0.09	0.15	0.09	0.12	0.07	0.10	0.06	0.08	0.06	0.07	0.05	0.07	0.05	0.06	0.05	0.06	0.05	0.06			
Precision	0.73	0.77	0.72	0.76	0.73	0.74	0.72	0.74	0.72	0.73	0.72	0.74	0.72	0.73	0.72	0.73	0.72	0.73	0.72	0.73	0.72	0.73			
Recall	0.82	0.66	0.87	0.81	0.91	0.85	0.91	0.88	0.93	0.90	0.94	0.92	0.94	0.93	0.95	0.93	0.95	0.95	0.94	0.95	0.94	0.95			
F1	0.77	0.71	0.79	0.78	0.81	0.79	0.80	0.80	0.81	0.81	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82			
il valore di k che porta precisione migliore è 2 che classifica 833.0 TP con precisione 0.77 con k=2, weight=uniform e p=1																									

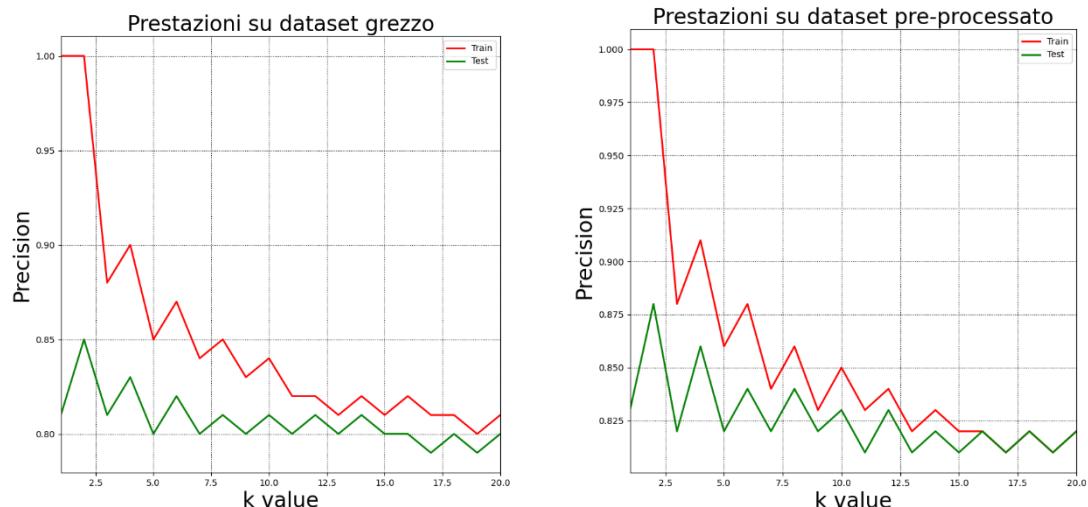
Probabilistico:

La terza strategia utilizzata è **probabilistica**, che mantiene i record della classe maggioritaria che hanno la maggior probabilità di essere classificati correttamente.

Nella selezione degli attributi i due plot mostrati sotto evidenziano la selezione oculata dei record, dato che vengono mantenuti quelli “al centro” mentre i record più vicini alla decision boundary sono esclusi perché hanno meno probabilità di essere classificati.

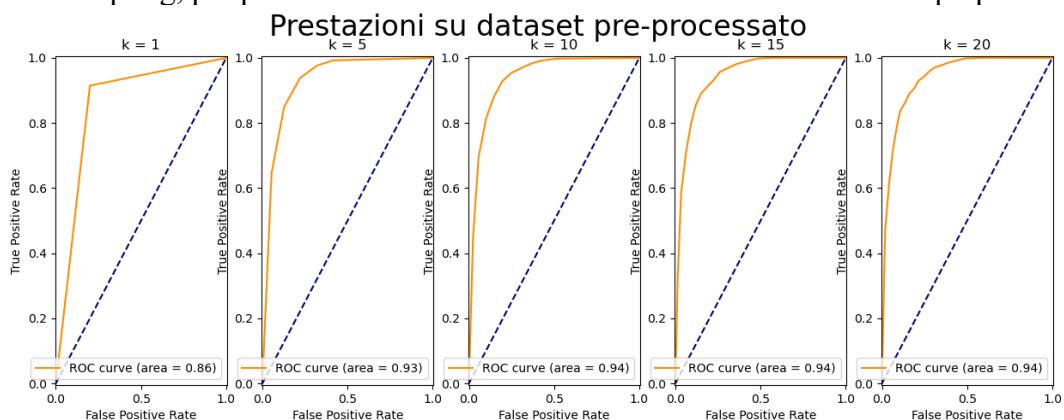


Usando una strategia di selezione più potente le prestazioni migliorano notevolmente, come si vede chiaramente dalle immagini.



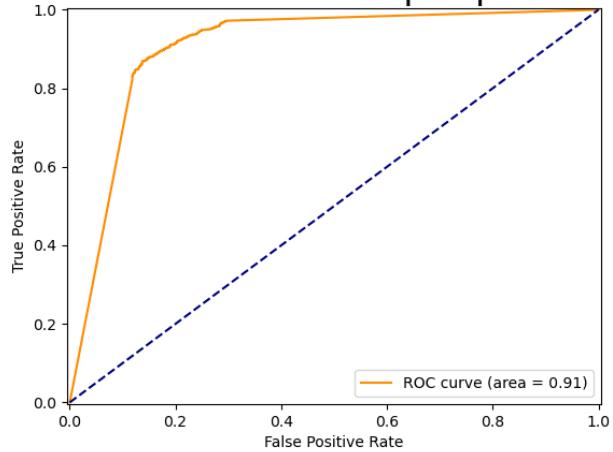
La precision migliora sia nel training set che nel test set sia rispetto al modello randomico sia rispetto al modello usato con i dati grezzi, grazie ai dati selezionati in maniera intelligente.

Il miglioramento è evidente anche nella curva ROC che presenta un valore di AUC molto più elevato rispetto alla curva ottenuta con i dati non processati. Infatti le prestazioni che si ottengono con $k=20$ nel modello con i dati iniziali sono le stesse che si ottengono con $k=1$ nel modello che analizza dati dopo l'undersampling, per poi ovviamente aumentare come cresce il valore dell'iperparametro k .



Prestazioni kNN tuned

Prestazioni su dataset pre-processato



Come nei due casi precedenti il valore di k migliore è pari a 2, mentre la precision aumenta fino a 0.88; anche stavolta come nei due algoritmi precedenti non sono state pesate le distanze ed è stata usata la distanza Manhattan. La curva rappresenta il miglior compromesso tra complessità e prestazioni del modello. Si tratta di un risultato considerevole, da tenere a mente per i migliori modelli.

Confronto per effettuare il tuning degli iperparametri modificando: iperparametro k (prima immagine), pesando le distanze (seconda immagine), e usando la distanza Manhattan invece di quella Euclidea (terza immagine). Infine si è selezionato il miglior valore (quarta immagine).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.86	0.85	0.86	0.87	0.86	0.87	0.86	0.87	0.86	0.87	0.86	0.86	0.86	0.85	0.86	0.86	0.86	0.86	0.86	
Error rate	0.14	0.15	0.14	0.13	0.14	0.13	0.14	0.13	0.14	0.13	0.14	0.14	0.14	0.15	0.14	0.14	0.14	0.14	0.14	
TP	1180.00	1060.00	1201.00	1146.00	1210.00	1170.00	1207.00	1187.00	1211.00	1197.00	1215.00	1195.00	1211.00	1191.00	1205.00	1197.00	1207.00	1199.00	1209.00	
TPR	0.91	0.82	0.93	0.89	0.94	0.91	0.94	0.92	0.94	0.93	0.94	0.93	0.94	0.92	0.93	0.93	0.94	0.93	0.94	0.93
TNR	0.80	0.88	0.79	0.85	0.78	0.82	0.78	0.81	0.78	0.80	0.77	0.80	0.77	0.79	0.77	0.79	0.78	0.78	0.77	0.79
FPR	0.20	0.12	0.21	0.15	0.22	0.18	0.22	0.19	0.22	0.20	0.23	0.20	0.23	0.21	0.23	0.21	0.22	0.22	0.23	0.21
FNR	0.09	0.18	0.07	0.11	0.06	0.09	0.06	0.08	0.06	0.07	0.06	0.07	0.06	0.08	0.07	0.07	0.06	0.07	0.06	0.07
Precision	0.83	0.88	0.82	0.86	0.82	0.84	0.82	0.84	0.82	0.83	0.81	0.83	0.81	0.82	0.81	0.82	0.81	0.82	0.81	0.82
Recall	0.91	0.82	0.93	0.89	0.94	0.91	0.94	0.92	0.94	0.93	0.94	0.93	0.94	0.92	0.93	0.93	0.94	0.93	0.94	0.93
F1	0.87	0.85	0.87	0.87	0.87	0.87	0.88	0.87	0.88	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
il valore di k che porta precisione migliore è 2 che classifica 1060.0 TP con precisione 0.88 senza cambiare alcun iperparametro eccetto k																				
Metriches di valutazione, test set:																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.86	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
Error rate	0.14	0.14	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
TP	1180.00	1180.00	1204.00	1221.00	1212.00	1214.00	1210.00	1217.00	1219.00	1222.00	1222.00	1222.00	1217.00	1218.00	1220.00	1219.00	1220.00	1219.00	1220.00	1221.00
TPR	0.91	0.91	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.94	0.94	0.95	0.94	0.95	0.94	0.95	0.95
TNR	0.80	0.80	0.80	0.80	0.79	0.79	0.79	0.79	0.79	0.78	0.79	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
FPR	0.20	0.20	0.20	0.20	0.21	0.21	0.21	0.21	0.21	0.21	0.22	0.21	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
FNR	0.09	0.09	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.06	0.05	0.06	0.05	0.06	0.05	0.06	0.05	0.05
Precision	0.83	0.83	0.83	0.83	0.83	0.82	0.83	0.83	0.83	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82
Recall	0.91	0.91	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.95	0.95	0.94	0.94	0.95	0.94	0.95	0.94	0.95	0.95	0.95
F1	0.87	0.87	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
il valore di k che porta precisione migliore è 1 che classifica 1180.0 TP con precisione 0.83 pesando le distanze																				
Metriches di valutazione, test set:																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.86	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
Error rate	0.14	0.14	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
TP	1192.00	1084.00	1218.00	1167.00	1223.00	1187.00	1231.00	1211.00	1233.00	1218.00	1235.00	1218.00	1235.00	1215.00	1228.00	1216.00	1231.00	1217.00	1228.00	1219.00
TPR	0.92	0.84	0.94	0.90	0.95	0.92	0.95	0.94	0.96	0.94	0.96	0.94	0.96	0.94	0.95	0.94	0.95	0.94	0.95	0.94
TNR	0.79	0.88	0.79	0.84	0.78	0.82	0.77	0.81	0.77	0.80	0.77	0.80	0.78	0.80	0.78	0.79	0.77	0.79	0.78	0.78
FPR	0.21	0.12	0.21	0.16	0.22	0.18	0.23	0.19	0.23	0.20	0.23	0.20	0.22	0.20	0.22	0.21	0.23	0.21	0.23	0.22
FNR	0.08	0.16	0.06	0.10	0.05	0.08	0.05	0.06	0.04	0.06	0.04	0.06	0.04	0.05	0.06	0.05	0.06	0.05	0.06	0.06
Precision	0.82	0.88	0.83	0.86	0.82	0.84	0.82	0.84	0.81	0.83	0.82	0.83	0.82	0.83	0.82	0.82	0.82	0.82	0.82	0.82
Recall	0.92	0.84	0.94	0.90	0.95	0.92	0.95	0.94	0.96	0.94	0.96	0.94	0.96	0.94	0.95	0.94	0.95	0.94	0.95	0.94
F1	0.87	0.86	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
il valore di k che porta precisione migliore è 2 che classifica 1084.0 TP con precisione 0.88 utilizzando la distanza Manhattan																				

```

Metriche di valutazione, test set:
      2
Accuracy      0.86
Error rate    0.14
TP            1084.00
TPR           0.84
TNR           0.88
FPR           0.12
FNR           0.16
Precision     0.88
Recall        0.84
F1            0.86
il valore di k che porta precisione migliore è 2 che classifica 1084.0 TP con precisione 0.88 con k=2, weight=uniform e p=1

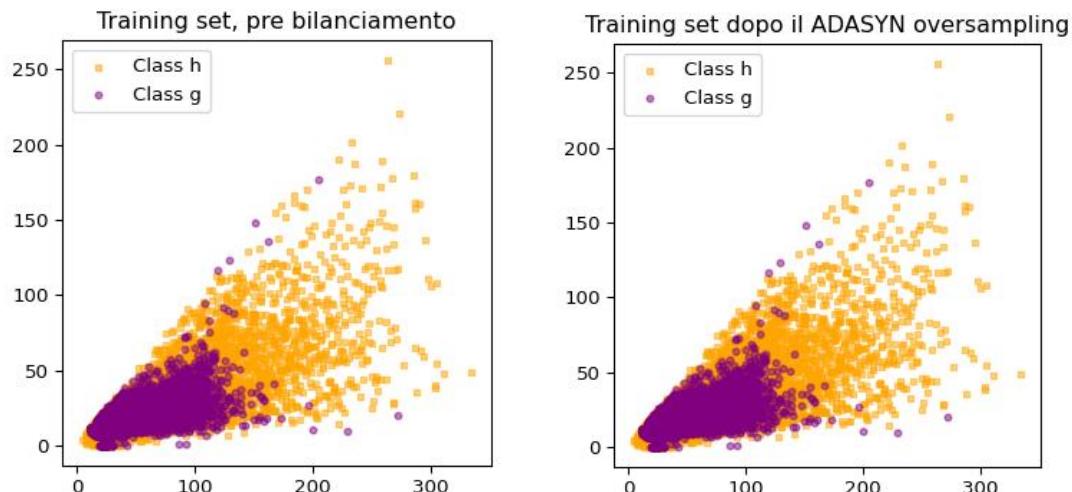
```

Oversampling

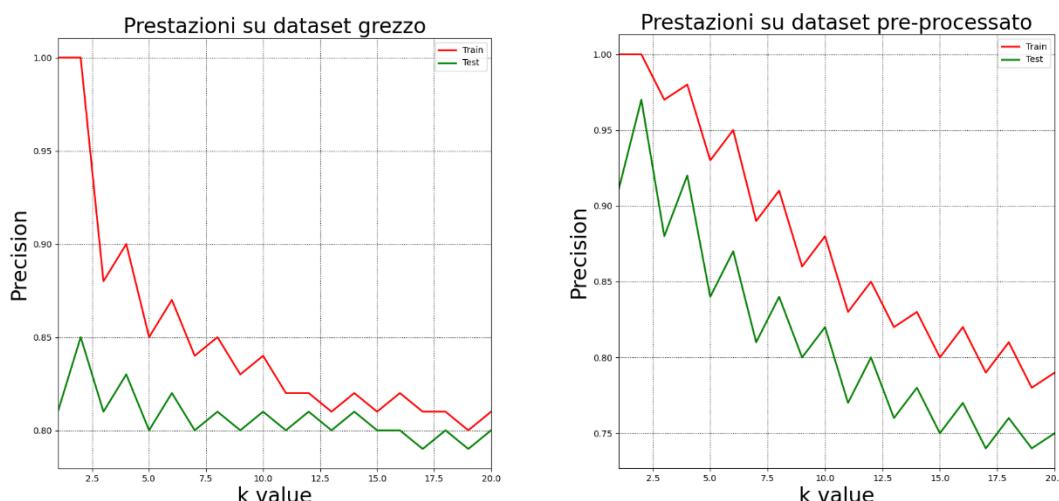
ADASYN:

Nell'oversampling sono state utilizzate tre differenti misure per aumentare il numero dei record della classe minoritaria. Come prima tecnica è stata utilizzata **ADASYN**, che si concentra sulla generazione di campioni vicini ai campioni originali classificati erroneamente con un classificatore kNN, migliorando la decision boundary.

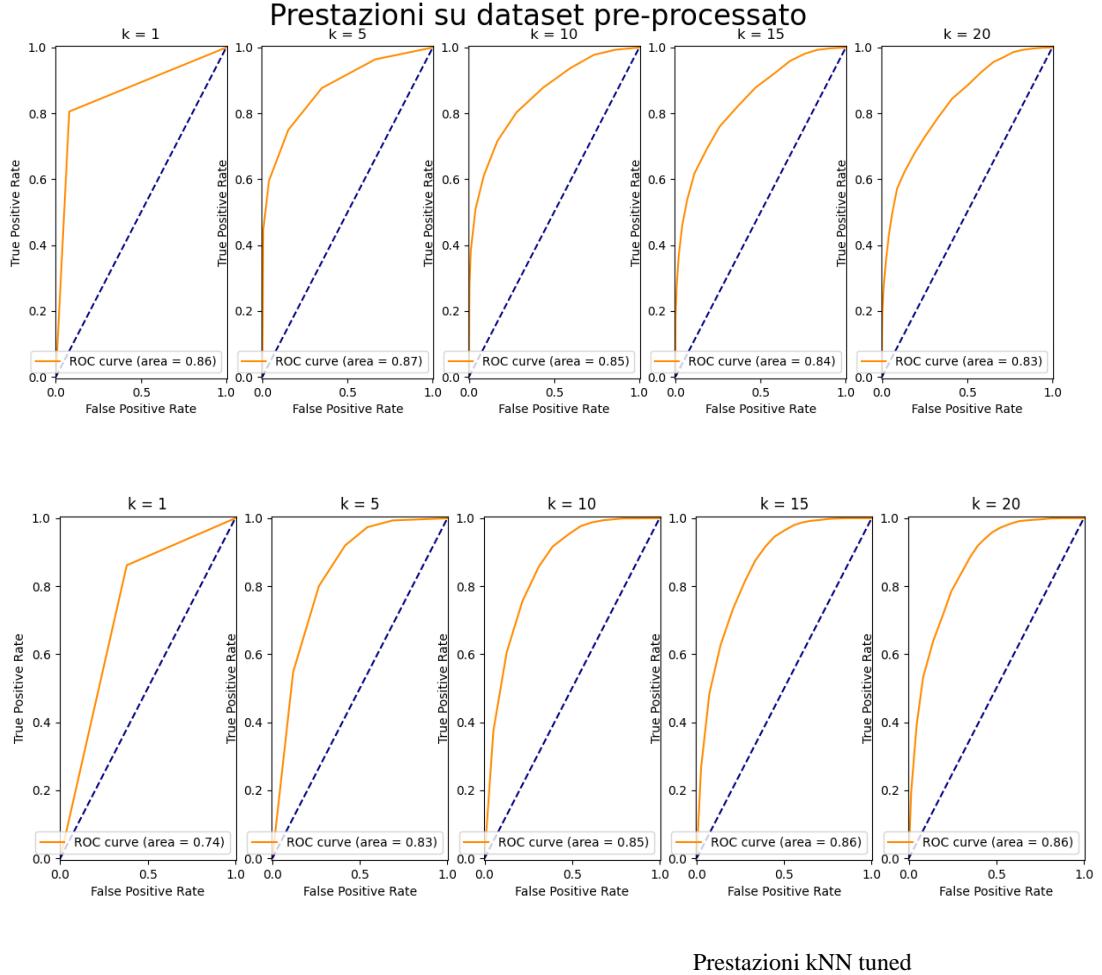
Per quanto riguarda la generazione di record h troviamo una distribuzione molto simile, questo perché è stata rinforzata la decision boundary vicino alla classe g .



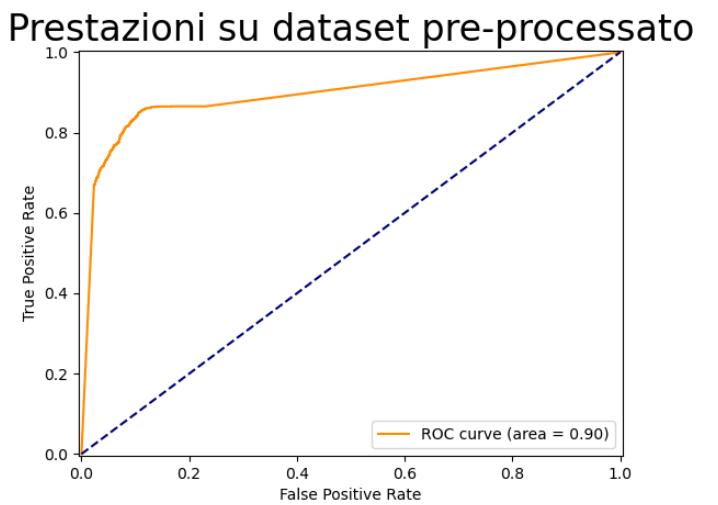
In questo caso le prestazioni sui dati grezzi, presenti sulla destra, risultano peggiori.



Osservando la curva ROC si può notare come le prestazioni sui dati pre-processati peggiorino rispetto alle prestazioni sui dati grezzi. L'eccezione è per la curva realizzata per $k=1$, nella quale l'area ha un valore superiore pari a 0,12. Questo è probabilmente dovuto al fatto che i dati risultano molto simili tra di loro e con l'incremento di k tra i vicini che si selezionano vi è una netta somiglianza, mentre nei dati grezzi ci sono valori più diversi tra di loro.



Per quanto riguarda l'oversampling, l'andamento della curva ROC per il kNN tuned, ($k=2$, senza pesare le distanze e usando distanza Manhattan) utilizzando il bilanciamento effettuato con ADASYN si presenta come il seguente. Questo è un risultato considerevole, l'AUC è 0,90, quasi la massima possibile. Per la precisione il risultato è ancora più considerevole: si tratta del maggiore visto finora: 0,97. Vengono classificati 1569 TP.



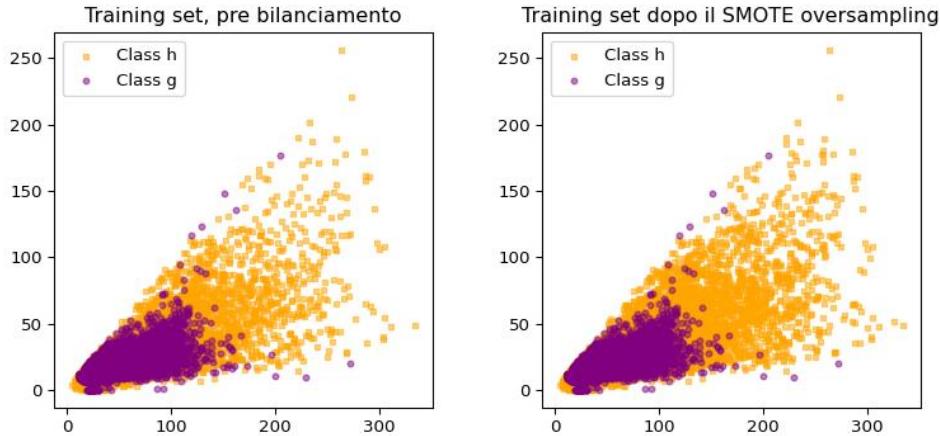
Metriche di valutazione kNN

Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy		0.86	0.82	0.82	0.80	0.80	0.79	0.79	0.78	0.78	0.77	0.76	0.76	0.75	0.75	0.75	0.75	0.74	0.74	0.74	0.74
Error rate		0.14	0.18	0.18	0.20	0.20	0.21	0.21	0.22	0.22	0.23	0.24	0.24	0.25	0.25	0.25	0.25	0.26	0.26	0.26	0.26
TP		1873.00	1550.00	1750.00	1539.00	1743.00	1593.00	1774.00	1640.00	1783.00	1665.00	1769.00	1667.00	1758.00	1672.00	1766.00	1691.00	1755.00	1692.00	1761.00	1698.00
TPR		0.81	0.67	0.75	0.66	0.75	0.69	0.76	0.71	0.77	0.72	0.76	0.72	0.76	0.72	0.76	0.75	0.75	0.73	0.75	0.73
TNR		0.92	0.98	0.90	0.94	0.85	0.89	0.82	0.86	0.79	0.83	0.77	0.81	0.75	0.79	0.74	0.77	0.72	0.76	0.72	0.75
FPR		0.08	0.02	0.10	0.06	0.15	0.11	0.18	0.14	0.21	0.17	0.23	0.19	0.25	0.21	0.26	0.23	0.28	0.24	0.28	0.25
FNR		0.19	0.33	0.25	0.34	0.25	0.31	0.24	0.29	0.23	0.28	0.24	0.28	0.24	0.28	0.24	0.27	0.25	0.27	0.24	0.27
Precision		0.91	0.97	0.88	0.92	0.84	0.87	0.81	0.84	0.80	0.82	0.77	0.80	0.76	0.78	0.75	0.77	0.74	0.76	0.74	0.75
Recall		0.81	0.67	0.75	0.66	0.75	0.69	0.76	0.71	0.77	0.72	0.76	0.72	0.76	0.72	0.76	0.73	0.75	0.73	0.76	0.73
F1		0.86	0.79	0.81	0.77	0.79	0.77	0.79	0.78	0.76	0.75	0.76	0.75	0.76	0.75	0.76	0.75	0.75	0.74	0.75	0.74
il valore di k che porta precisione migliore è 2 che classifica 1550.0 TP con precisione 0.97 senza cambiare alcun iperparametro eccetto k																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy		0.86	0.86	0.84	0.85	0.83	0.84	0.84	0.84	0.84	0.83	0.83	0.83	0.82	0.82	0.82	0.82	0.81	0.81	0.81	0.81
Error rate		0.14	0.14	0.16	0.15	0.17	0.16	0.16	0.16	0.16	0.17	0.17	0.17	0.18	0.18	0.18	0.18	0.19	0.19	0.19	0.19
TP		1873.00	1873.00	1762.00	1785.00	1755.00	1786.00	1792.00	1799.00	1800.00	1800.00	1786.00	1797.00	1782.00	1786.00	1787.00	1787.00	1782.00	1790.00	1785.00	1787.00
TPR		0.81	0.81	0.76	0.77	0.75	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
TNR		0.92	0.92	0.93	0.93	0.92	0.92	0.91	0.91	0.90	0.89	0.89	0.88	0.87	0.87	0.87	0.86	0.86	0.85	0.85	0.84
FPR		0.08	0.08	0.07	0.07	0.08	0.08	0.09	0.09	0.10	0.11	0.11	0.12	0.13	0.13	0.14	0.14	0.15	0.15	0.16	0.16
FNR		0.19	0.19	0.24	0.23	0.25	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23
Precision		0.91	0.91	0.92	0.92	0.91	0.91	0.90	0.90	0.89	0.88	0.88	0.87	0.86	0.86	0.85	0.85	0.84	0.84	0.84	0.84
Recall		0.81	0.81	0.76	0.77	0.75	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
F1		0.86	0.86	0.83	0.84	0.82	0.83	0.83	0.83	0.83	0.82	0.82	0.82	0.81	0.81	0.81	0.81	0.81	0.80	0.80	0.80
il valore di k che porta precisione migliore è 3 che classifica 1762.0 TP con precisione 0.92 pesando le distanze																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy		0.86	0.82	0.83	0.82	0.81	0.81	0.80	0.79	0.78	0.78	0.77	0.78	0.77	0.77	0.76	0.76	0.75	0.76	0.75	0.75
Error rate		0.14	0.18	0.17	0.18	0.19	0.19	0.20	0.21	0.22	0.22	0.23	0.22	0.23	0.24	0.24	0.25	0.24	0.25	0.25	0.25
TP		1879.00	1569.00	1800.00	1617.00	1808.00	1661.00	1820.00	1702.00	1817.00	1713.00	1828.00	1740.00	1832.00	1755.00	1841.00	1750.00	1822.00	1759.00	1827.00	1766.00
TPR		0.81	0.67	0.77	0.70	0.78	0.71	0.78	0.73	0.78	0.74	0.79	0.75	0.79	0.75	0.79	0.75	0.78	0.76	0.79	0.76
TNR		0.92	0.98	0.90	0.94	0.84	0.90	0.82	0.85	0.78	0.83	0.76	0.81	0.75	0.79	0.74	0.77	0.72	0.76	0.71	0.75
FPR		0.08	0.02	0.10	0.06	0.16	0.10	0.18	0.15	0.22	0.17	0.24	0.19	0.25	0.21	0.26	0.23	0.28	0.24	0.29	0.25
FNR		0.19	0.33	0.23	0.30	0.22	0.29	0.22	0.27	0.22	0.26	0.21	0.25	0.21	0.25	0.22	0.24	0.21	0.22	0.24	0.24
Precision		0.91	0.97	0.89	0.92	0.84	0.88	0.82	0.84	0.79	0.82	0.77	0.80	0.77	0.79	0.76	0.77	0.74	0.77	0.74	0.76
Recall		0.81	0.67	0.77	0.70	0.78	0.71	0.78	0.73	0.78	0.74	0.79	0.75	0.79	0.75	0.78	0.76	0.78	0.76	0.79	0.76
F1		0.86	0.79	0.83	0.79	0.81	0.79	0.80	0.78	0.79	0.78	0.78	0.77	0.78	0.77	0.77	0.76	0.76	0.76	0.76	0.76
il valore di k che porta precisione migliore è 2 che classifica 1569.0 TP con precisione 0.97 utilizzando la distanza Manhattan																					
Metriche di valutazione, test set:		2																			
Accuracy		0.82																			
Error rate		0.18																			
TP		1569.00																			
TPR		0.67																			
TNR		0.98																			
FPR		0.02																			
FNR		0.33																			
Precision		0.97																			
Recall		0.67																			
F1		0.79																			
il valore di k che porta precisione migliore è 2 che classifica 1569.0 TP con precisione 0.97 con k=2, weight=uniform e p=1																					

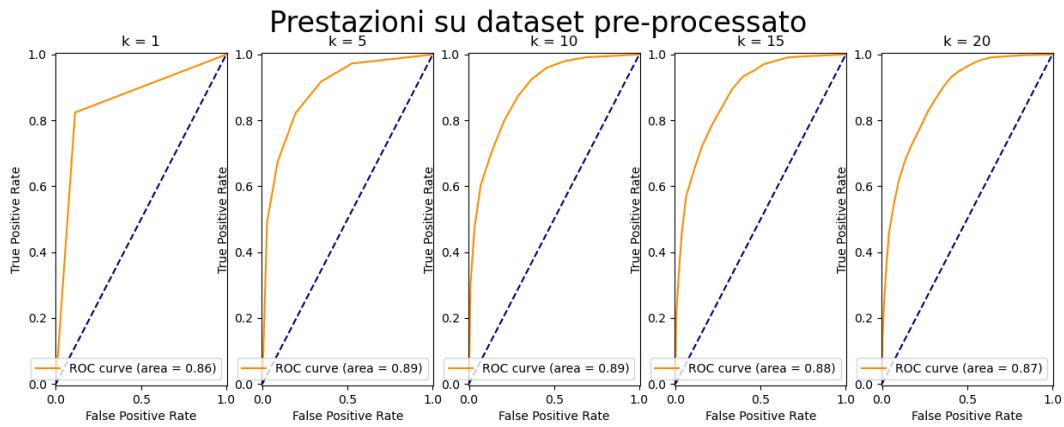
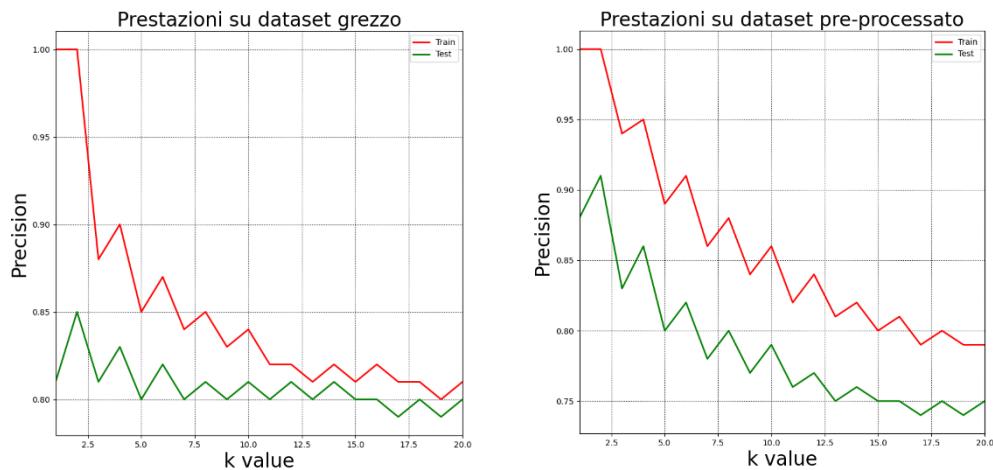
SMOTE:

Il secondo algoritmo utilizzato è **SMOTE**, che produce indistintamente campioni sfruttando la regola dei vicini.

Nella rappresentazione grafica, a differenza del plot precedente mostrato con ADASYN, si nota come la distribuzione dei nuovi record generati avvenga per tutta la superficie coperta dai record h e non solo lungo la decision boundary.

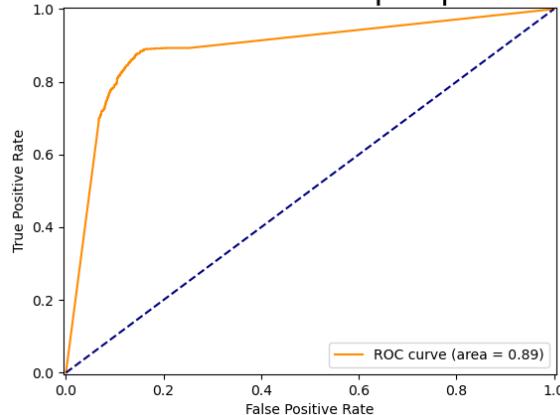


Le prestazioni sono nettamente migliori rispetto ai dati grezzi come si può notare sia per la curva ROC, in quanto l'area sottostante la curva aumenta, sia per il grafico della *precision*.



Prestazioni kNN tuned

Prestazioni su dataset pre-processato



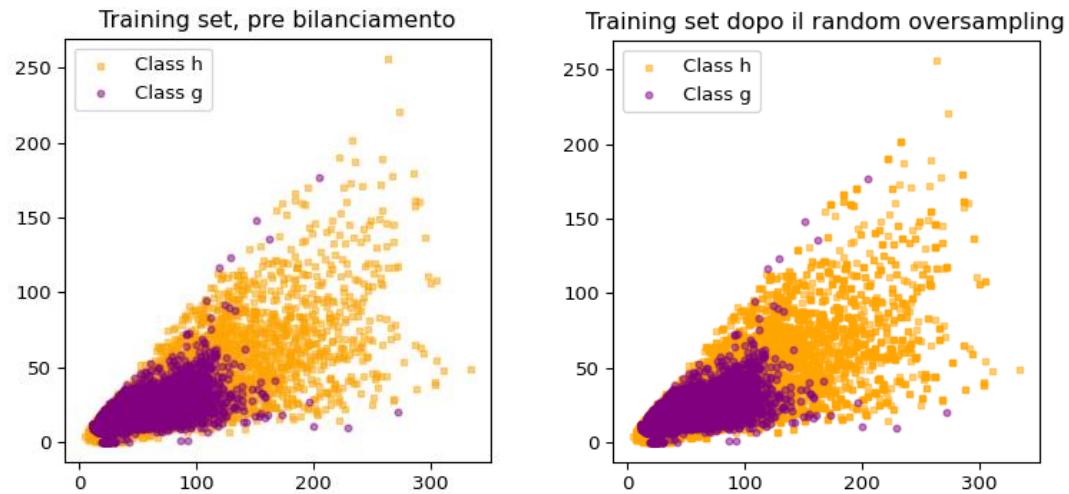
A sinistra è riportato il grafico della curva ROC relativo all'oversampling con SMOTE, per quanto riguarda il kNN tuned. L'area si presenta ottima ma leggermente minore rispetto alle ultime. La precisione si assesta a un ottimo 0.91.

Metriche di valutazione kNN

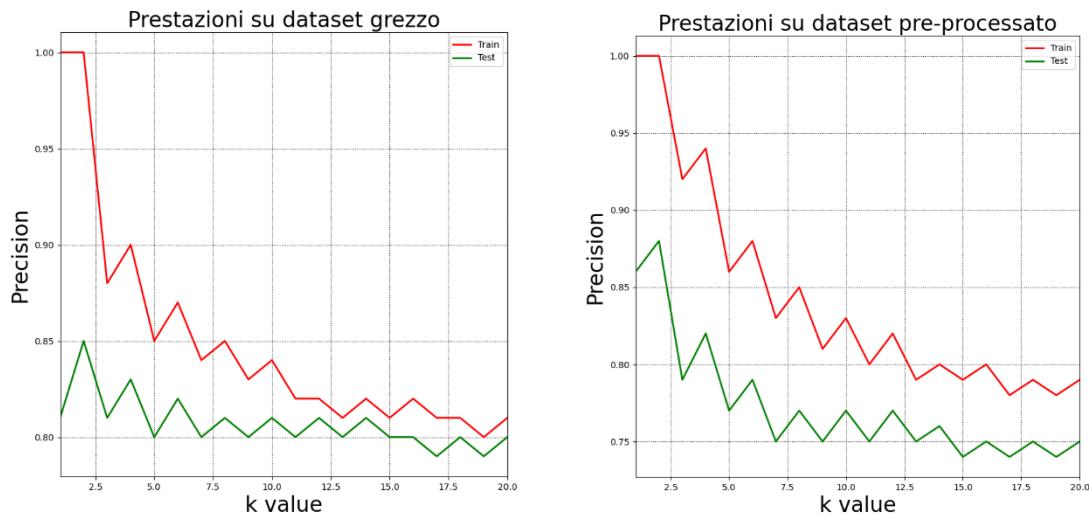
Metriche di valutazione, test set:	
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Accuracy	0.86 0.81 0.82 0.81 0.81 0.81 0.80 0.80 0.80 0.80 0.79 0.79 0.78 0.78 0.78 0.78 0.78 0.78 0.78 0.78
Error rate	0.14 0.19 0.18 0.19 0.19 0.19 0.20 0.20 0.20 0.20 0.21 0.21 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.22
TP	1871.00 1563.00 1825.00 1665.00 1868.00 1749.00 1879.00 1789.00 1901.00 1811.00 1907.00 1832.00 1905.00 1851.00 1917.00 1866.00 1922.00 1871.00 1929.00 1881.00
TPR	0.82 0.69 0.80 0.73 0.82 0.77 0.83 0.79 0.84 0.80 0.84 0.81 0.84 0.82 0.84 0.82 0.85 0.82 0.85 0.85 0.83
TNR	0.89 0.93 0.84 0.89 0.80 0.84 0.77 0.81 0.76 0.79 0.74 0.77 0.73 0.75 0.72 0.74 0.71 0.73 0.71 0.73 0.73
FPR	0.11 0.07 0.16 0.11 0.20 0.16 0.23 0.19 0.24 0.21 0.26 0.23 0.27 0.25 0.28 0.26 0.29 0.27 0.27 0.29 0.27
FNR	0.18 0.31 0.20 0.27 0.18 0.23 0.17 0.21 0.16 0.20 0.16 0.19 0.16 0.18 0.16 0.18 0.15 0.18 0.15 0.17 0.17
Precision	0.88 0.91 0.85 0.86 0.88 0.82 0.78 0.80 0.77 0.79 0.76 0.77 0.75 0.76 0.75 0.75 0.74 0.75 0.74 0.75 0.75
Recall	0.82 0.69 0.80 0.73 0.82 0.77 0.83 0.79 0.84 0.80 0.84 0.81 0.84 0.82 0.84 0.82 0.85 0.82 0.85 0.85 0.83
F1	0.85 0.78 0.82 0.79 0.81 0.80 0.80 0.79 0.80 0.79 0.80 0.79 0.79 0.79 0.79 0.79 0.79 0.79 0.78 0.79 0.79
il valore di k che porta precisione migliore è 2 che classifica 1563.0 TP con precisione 0.91 senza cambiare alcun iperparametro eccetto k	
Metriche di valutazione, test set:	
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Accuracy	0.86 0.86 0.85 0.85 0.84 0.84 0.84 0.84 0.84 0.84 0.84 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.82
Error rate	0.14 0.14 0.15 0.15 0.16 0.16 0.16 0.16 0.16 0.16 0.16 0.17 0.17 0.17 0.17 0.17 0.17 0.17 0.17 0.17 0.18
TP	1871.00 1871.00 1833.00 1854.00 1876.00 1886.00 1891.00 1897.00 1910.00 1905.00 1914.00 1920.00 1925.00 1931.00 1933.00 1939.00 1932.00 1937.00 1944.00 1940.00
TPR	0.82 0.82 0.81 0.82 0.83 0.83 0.83 0.84 0.84 0.84 0.84 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85
TNR	0.89 0.89 0.88 0.88 0.86 0.86 0.86 0.85 0.84 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83
FPR	0.11 0.11 0.12 0.12 0.14 0.14 0.15 0.15 0.16 0.16 0.17 0.17 0.17 0.17 0.17 0.19 0.19 0.19 0.19 0.20 0.21
FNR	0.18 0.18 0.19 0.18 0.17 0.17 0.17 0.16 0.16 0.16 0.16 0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15
Precision	0.88 0.88 0.87 0.86 0.85 0.85 0.84 0.84 0.83 0.83 0.83 0.82 0.82 0.82 0.82 0.81 0.81 0.80 0.80 0.80 0.80
Recall	0.82 0.82 0.81 0.82 0.83 0.83 0.83 0.84 0.84 0.84 0.84 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.86 0.85
F1	0.85 0.85 0.84 0.84 0.84 0.84 0.84 0.84 0.84 0.84 0.84 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83
il valore di k che porta precisione migliore è 1 che classifica 1871.0 TP con precisione 0.88 pesando le distanze	
Metriche di valutazione, test set:	
	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Accuracy	0.85 0.82 0.83 0.82 0.82 0.81 0.81 0.81 0.80 0.80 0.80 0.79 0.79 0.79 0.79 0.79 0.79 0.79 0.79 0.79 0.79
Error rate	0.15 0.18 0.17 0.18 0.18 0.18 0.19 0.19 0.19 0.20 0.20 0.20 0.21 0.21 0.21 0.21 0.21 0.21 0.21 0.21 0.21
TP	1865.00 1599.00 1868.00 1701.00 1889.00 1764.00 1914.00 1829.00 1942.00 1871.00 1957.00 1905.00 1965.00 1909.00 1967.00 1912.00 1968.00 1926.00 1984.00 1936.00
TPR	0.82 0.70 0.82 0.75 0.83 0.78 0.84 0.81 0.86 0.82 0.86 0.84 0.87 0.84 0.87 0.84 0.87 0.84 0.87 0.85 0.85
TNR	0.88 0.93 0.83 0.88 0.80 0.84 0.77 0.81 0.76 0.79 0.74 0.77 0.72 0.75 0.72 0.74 0.71 0.73 0.70 0.72
FPR	0.12 0.07 0.17 0.12 0.20 0.16 0.23 0.19 0.24 0.21 0.26 0.23 0.28 0.25 0.28 0.26 0.29 0.27 0.30 0.28
FNR	0.18 0.30 0.18 0.25 0.17 0.22 0.16 0.19 0.14 0.18 0.14 0.16 0.13 0.16 0.13 0.16 0.13 0.15 0.13 0.15
Precision	0.87 0.91 0.83 0.86 0.80 0.82 0.78 0.80 0.77 0.79 0.76 0.78 0.75 0.76 0.75 0.76 0.74 0.75 0.74 0.75 0.75
Recall	0.82 0.70 0.82 0.75 0.83 0.78 0.84 0.81 0.86 0.82 0.86 0.84 0.87 0.84 0.87 0.84 0.87 0.85 0.87 0.85 0.85
F1	0.85 0.79 0.82 0.80 0.82 0.80 0.81 0.80 0.81 0.81 0.81 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83
il valore di k che porta precisione migliore è 2 che classifica 1599.0 TP con precisione 0.91 utilizzando la distanza Manhattan	
Metriche di valutazione, test set:	
	2
Accuracy	0.82
Error rate	0.18
TP	1599.00
TPR	0.70
TNR	0.93
FPR	0.07
FNR	0.30
Precision	0.91
Recall	0.70
F1	0.79
il valore di k che porta precisione migliore è 2 che classifica 1599.0 TP con precisione 0.91 con k=2, weight=uniform e p=1	

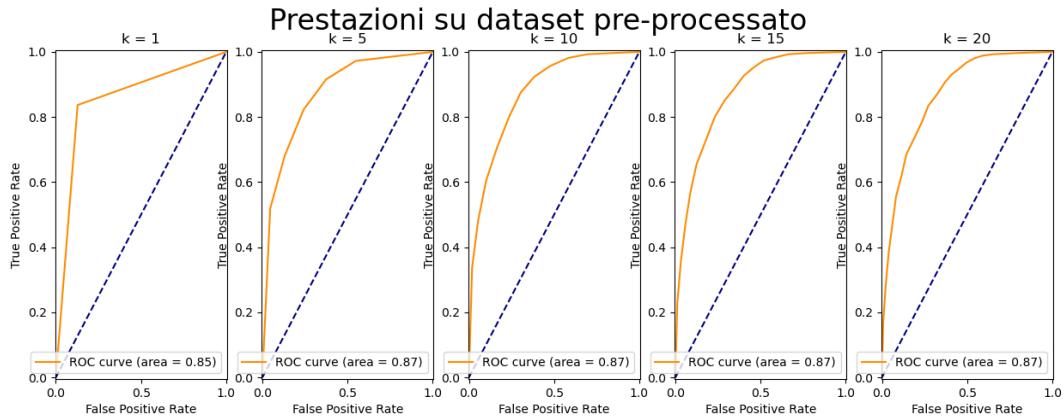
RandomOverSampler:

La terza misura utilizzata è il **RandomOverSampler**, nel quale gli elementi della classe minoritaria vengono generati in maniera randomica.

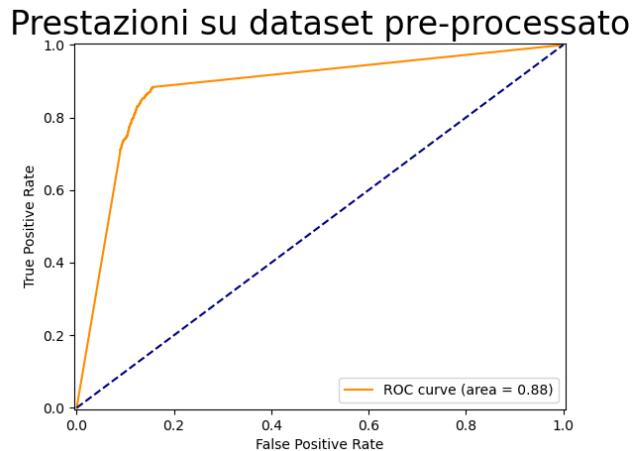


Anche in questo caso, come le strategie precedenti, le prestazioni sul dataset processato sono decisamente migliori rispetto al dataset grezzo non trattato. Tali considerazioni sono osservabili sia sul grafico della *precision* sia sulla curva ROC.





Prestazioni kNN tuned



Di seguito è rappresentata la curva ROC trovata con i valori del kNN tuned. L'area è alta in linea coi risultati delle varie tecniche di oversampling. Si arriva ad una precisione di 0.89.

Metrichi di valutazione kNN																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Accuracy	0.85	0.81	0.80	0.79	0.79	0.79	0.78	0.78	0.79	0.78	0.79	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	
Error rate	0.15	0.19	0.20	0.21	0.21	0.21	0.22	0.22	0.21	0.22	0.21	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	
TP	1900.00	1596.00	1823.00	1661.00	1868.00	1756.00	1881.00	1788.00	1907.00	1816.00	1913.00	1842.00	1919.00	1869.00	1933.00	1887.00	1935.00	1876.00	1932.00	1895.00	
TPR	0.84	0.70	0.80	0.73	0.82	0.77	0.83	0.79	0.84	0.80	0.84	0.81	0.85	0.82	0.85	0.83	0.85	0.83	0.85	0.85	0.83
TNR	0.87	0.91	0.79	0.84	0.76	0.80	0.74	0.78	0.73	0.76	0.73	0.76	0.72	0.75	0.71	0.74	0.71	0.73	0.71	0.73	0.73
FPR	0.13	0.09	0.21	0.16	0.24	0.20	0.26	0.22	0.27	0.24	0.27	0.24	0.28	0.25	0.29	0.26	0.29	0.27	0.29	0.27	0.27
FNR	0.16	0.30	0.20	0.27	0.18	0.23	0.17	0.21	0.16	0.20	0.16	0.19	0.15	0.18	0.15	0.17	0.15	0.17	0.15	0.15	0.17
Precision	0.86	0.88	0.79	0.82	0.77	0.79	0.75	0.77	0.75	0.77	0.75	0.77	0.75	0.76	0.74	0.75	0.74	0.75	0.74	0.75	0.75
Recall	0.84	0.70	0.80	0.73	0.82	0.77	0.83	0.79	0.84	0.80	0.84	0.81	0.85	0.82	0.85	0.83	0.85	0.83	0.85	0.85	0.83
F1	0.85	0.78	0.79	0.77	0.79	0.78	0.79	0.78	0.79	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
il valore di k che porta precisione migliore è 2 che classifica 1596.0 TP con precisione 0.88 senza cambiare alcun iperparametro eccetto k																					
Metrichi di valutazione, test set:																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Accuracy	0.85	0.85	0.85	0.86	0.86	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87	0.88	0.87	0.87	0.87	0.87	0.87	0.87	0.87
Error rate	0.15	0.15	0.15	0.14	0.14	0.14	0.14	0.14	0.15	0.13	0.13	0.13	0.13	0.12	0.13	0.13	0.13	0.13	0.13	0.13	0.13
TP	1900.00	1900.00	1831.00	1871.00	1873.00	1895.00	1894.00	1896.00	1909.00	1912.00	1926.00	1932.00	1935.00	1933.00	1945.00	1944.00	1944.00	1937.00	1946.00	1942.00	
TPR	0.84	0.84	0.81	0.82	0.82	0.83	0.83	0.83	0.84	0.84	0.85	0.85	0.85	0.85	0.86	0.86	0.86	0.85	0.86	0.86	0.86
TNR	0.87	0.87	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.90	0.89	0.89	0.89	0.89	0.89	0.89
FPR	0.13	0.13	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.10	0.11	0.11	0.11	0.11	0.11	0.11
FNR	0.16	0.16	0.19	0.18	0.18	0.17	0.17	0.17	0.16	0.16	0.15	0.15	0.15	0.15	0.14	0.14	0.14	0.14	0.15	0.14	0.14
Precision	0.86	0.86	0.87	0.87	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.89	0.89	0.89	0.88	0.88	0.88	0.88	0.88	0.88
Recall	0.84	0.84	0.81	0.82	0.82	0.83	0.83	0.84	0.84	0.84	0.85	0.85	0.85	0.86	0.86	0.86	0.85	0.86	0.86	0.86	0.86
F1	0.85	0.85	0.84	0.85	0.85	0.85	0.86	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
il valore di k che porta precisione migliore è 13 che classifica 1935.0 TP con precisione 0.89 pesando le distanze																					

Metriche di valutazione, test set:																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.85	0.82	0.79	0.79	0.79	0.79	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.78	0.78	0.78	0.78	0.78	0.78	0.78
Error rate	0.15	0.18	0.21	0.21	0.21	0.21	0.22	0.21	0.21	0.21	0.21	0.21	0.21	0.22	0.22	0.22	0.22	0.22	0.22	0.22
TP	1913.00	1648.00	1848.00	1692.00	1900.00	1783.00	1908.00	1819.00	1933.00	1866.00	1950.00	1893.00	1958.00	1915.00	1966.00	1921.00	1974.00	1932.00	1982.00	1942.00
TPR	0.84	0.73	0.81	0.75	0.84	0.79	0.84	0.80	0.85	0.82	0.86	0.83	0.86	0.84	0.87	0.85	0.87	0.85	0.87	0.86
TNR	0.86	0.91	0.78	0.83	0.75	0.80	0.73	0.77	0.73	0.76	0.72	0.75	0.71	0.74	0.70	0.73	0.78	0.72	0.69	0.72
FPR	0.14	0.09	0.22	0.17	0.25	0.20	0.27	0.23	0.27	0.24	0.28	0.25	0.29	0.26	0.30	0.27	0.30	0.28	0.31	0.28
FNR	0.16	0.27	0.19	0.25	0.16	0.21	0.16	0.20	0.15	0.18	0.14	0.17	0.14	0.16	0.13	0.15	0.13	0.15	0.13	0.14
Precision	0.86	0.89	0.78	0.81	0.76	0.79	0.75	0.77	0.75	0.77	0.75	0.76	0.74	0.76	0.74	0.75	0.74	0.75	0.73	0.74
Recall	0.84	0.73	0.81	0.75	0.84	0.79	0.84	0.80	0.85	0.82	0.86	0.83	0.86	0.84	0.87	0.85	0.87	0.85	0.87	0.86
F1	0.85	0.80	0.80	0.78	0.80	0.79	0.79	0.78	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.79	0.80	0.80	0.80	0.80

Metriche di valutazione, test set:																				
	2																			
Accuracy	0.82																			
Error rate	0.18																			
TP	1648.00																			
TPR	0.73																			
TNR	0.91																			
FPR	0.09																			
FNR	0.27																			
Precision	0.89																			
Recall	0.73																			
F1	0.80																			

il valore di k che porta precisione migliore è 2 che classifica 1648.0 TP con precisione 0.89 utilizzando la distanza Manhattan

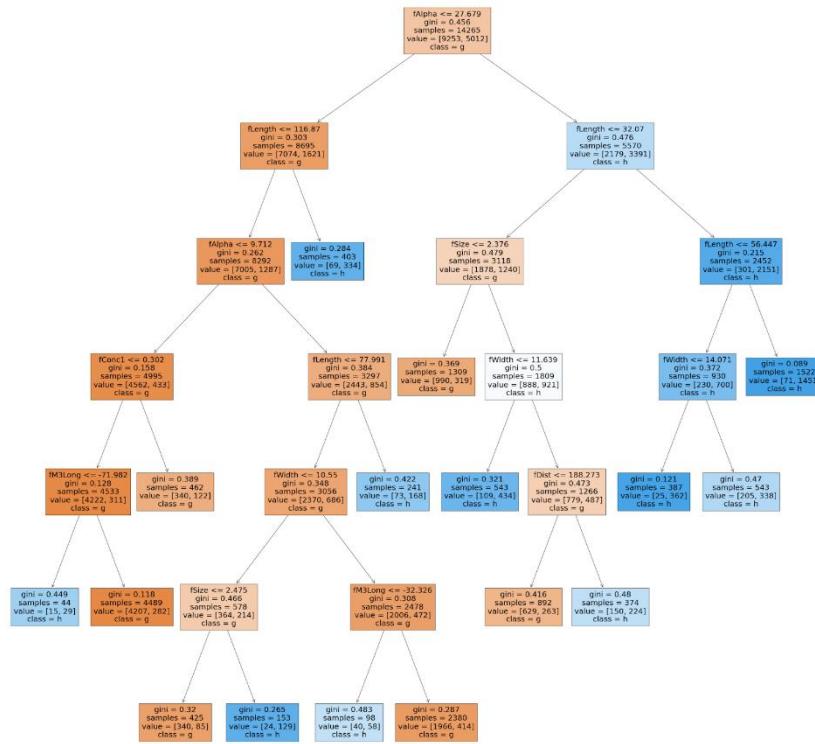
Come è stato dimostrato le prestazioni basate sulla precision nei dati processati con oversampling risultano nel complesso *estremamente* migliori rispetto a quelle ottenute con i dati grezzi e con i dati processati con undersampling. Questo è dovuto alla natura del dataset, il quale nella descrizione rende nota una sottostima dei campioni *h* rispetto a quelli *g*, che sono quelli della nostra classe di interesse. La diminuzione del numero di *g* comporta una perdita di dati preziosi utili alla classificazione, mentre l'aumento sintetico dei valori di *h* crea un dataset più veritiero. Non essendo presente uno sbilanciamento eccessivo l'oversampling non provoca un problema di overfitting verso la classe *h*, della quale si realizzano appunto molti record simili tra di loro.

In generale si nota che le prestazioni relative alla precision e all'AUC migliorano con l'oversampling piuttosto che con l'undersampling, in particolare con ADASYN si trova il miglior valore di precision per k=2, senza pesare le distanze ed utilizzando la distanza Manhattan, addirittura 0.97. La classificazione di un elemento come *g* è quasi perfetta. Essa ha un valore di AUC molto elevato (0.90), anche se non è quello massimo che è quello dell'undersampling probabilistico (0.91).

Classificatore logico: alberi decisionali

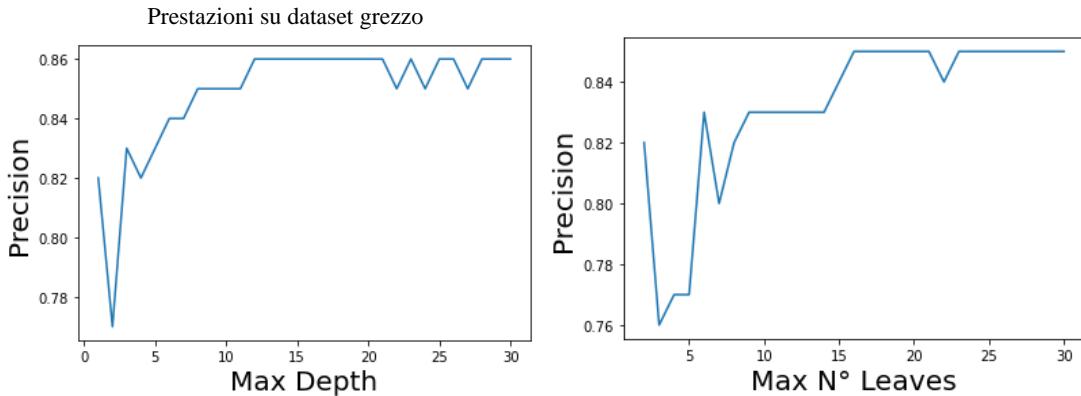
Sono state condotte le stesse operazioni attuate sui dati grezzi, ma il commento dei risultati si concentrerà sui tre albero ritenuti migliori: l'albero tuned, che massimizza i parametri max_depth e min_leaf_nodes sulla base della precisione, l'albero ottenuto con min gain 0.05 e l'albero ottenuto con min gain 0.005. E' da notare che questi alberi potrebbero avere caratteristiche molto diverse da quelle che li hanno portati ad essere le scelte migliori basate sui dati grezzi. Quando questi cambiamenti saranno significativi verranno evidenziati gli alberi migliori. Per ogni tecnica verranno mostrati i grafici che mostrano l'andamento della precisione al variare degli iperparametri, l'albero generato che li massimizza e gli alberi generati con i vari valori di min gain. Sono inoltre presenti le tabelle con le relative metriche. Viene ripetuta l'immagine dell'albero tuned del dataset grezzo per confronto.

Albero tuned dataset grezzo

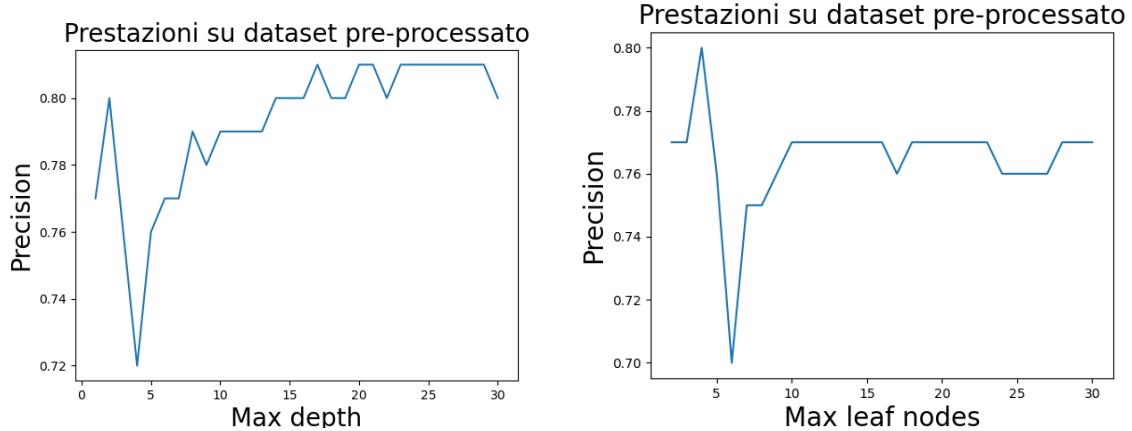


Undersampling

RandomUnderSampler:



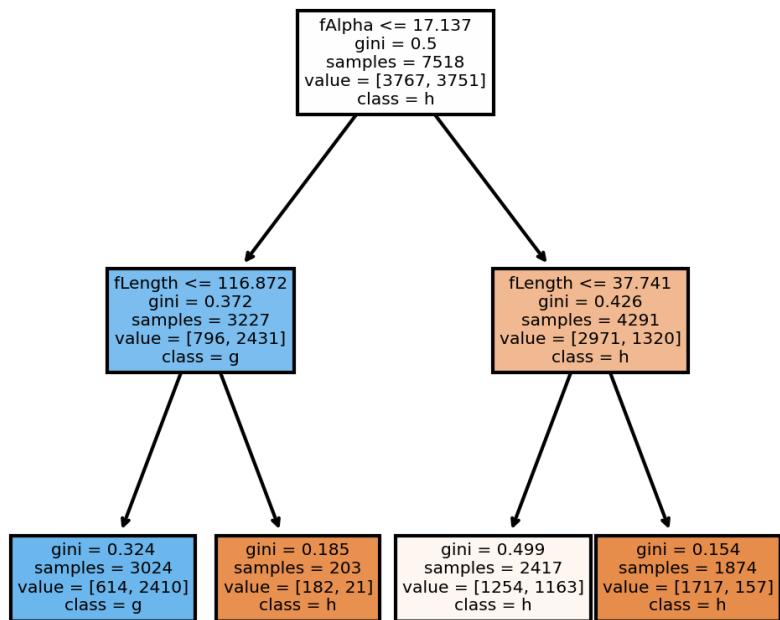
Prestazioni su dataset grezzo



Metriche di valutazione dTree

Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:						
Tuned						
Accuracy	0.74					
Error rate	0.26					
TP	796.00					
TPR	0.63					
TNR	0.84					
FPR	0.16					
FNR	0.37					
Precision	0.80					
Recall	0.63					
F1	0.71					
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:						
	0.1000	0.0500	0.0300	0.0080	0.0075	0.0050
Accuracy	0.5	0.72	0.72	0.76	0.78	0.79
Error rate	0.5	0.28	0.28	0.24	0.22	0.21
TP	0.0	807.00	807.00	1157.00	1085.00	1070.00
TPR	0.0	0.64	0.64	0.92	0.86	0.85
TNR	1.0	0.81	0.81	0.61	0.71	0.72
FPR	0.0	0.19	0.19	0.39	0.29	0.28
FNR	1.0	0.36	0.36	0.08	0.14	0.15
Precision	0.0	0.77	0.77	0.70	0.75	0.76
Recall	0.0	0.64	0.64	0.92	0.86	0.85
F1	0.0	0.70	0.70	0.80	0.80	0.80
Metriche di valutazione per il test set, con albero tuned, su dataset grezzo:						
	Tuned					
Accuracy	0.83					
Error rate	0.17					
TP	2796.00					
TPR	0.91					
TNR	0.70					
FPR	0.30					
FNR	0.09					
Precision	0.85					
Recall	0.91					
F1	0.88					
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset grezzo:						
	0.1000	0.0500	0.0300	0.0080	0.0075	0.0050
Accuracy	0.65	0.74	0.78	0.79	0.82	0.82
Error rate	0.35	0.26	0.22	0.21	0.18	0.18
TP	3079.00	2392.00	2984.00	2963.00	2930.00	2885.00
TPR	1.00	0.78	0.97	0.96	0.95	0.94
TNR	0.00	0.68	0.43	0.48	0.57	0.61
FPR	1.00	0.32	0.57	0.52	0.43	0.39
FNR	0.00	0.22	0.03	0.04	0.05	0.06
Precision	0.65	0.82	0.76	0.77	0.80	0.82
Recall	1.00	0.78	0.97	0.96	0.95	0.94
F1	0.79	0.80	0.85	0.86	0.87	0.87

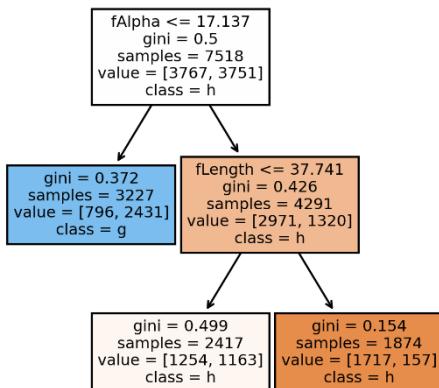
Albero tuned dataset processato



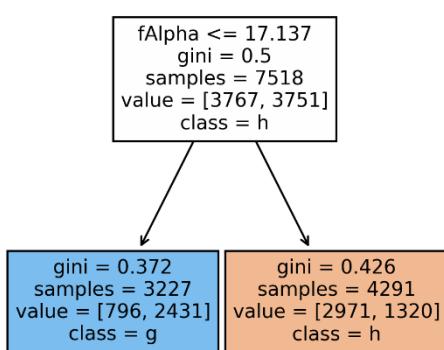
Tree pre-processato con mingain 0.1

**gini = 0.5
samples = 7518
value = [3767, 3751]
class = h**

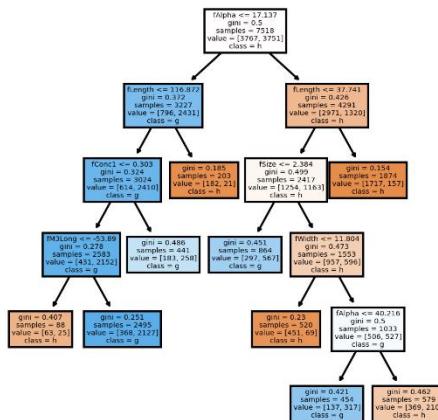
Tree pre-processato con mingain 0.03



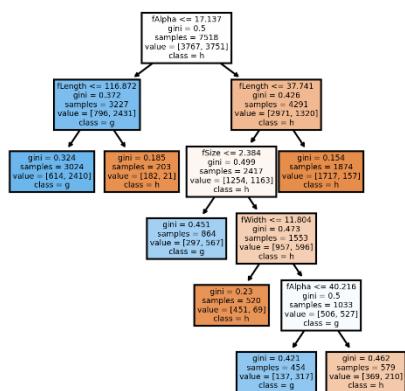
Tree pre-processato con mingain 0.05



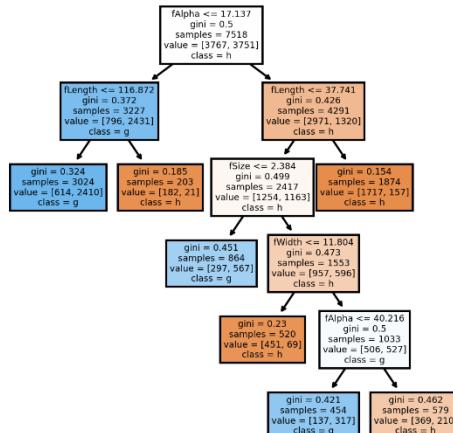
Tree pre-processato con mingain 0.005



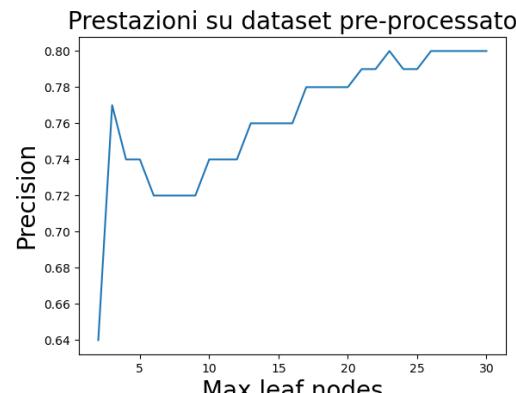
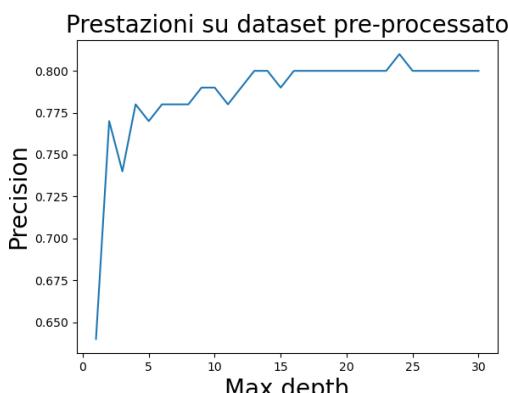
Tree pre-processato con mingain 0.008



Tree pre-processato con mingain 0.0075



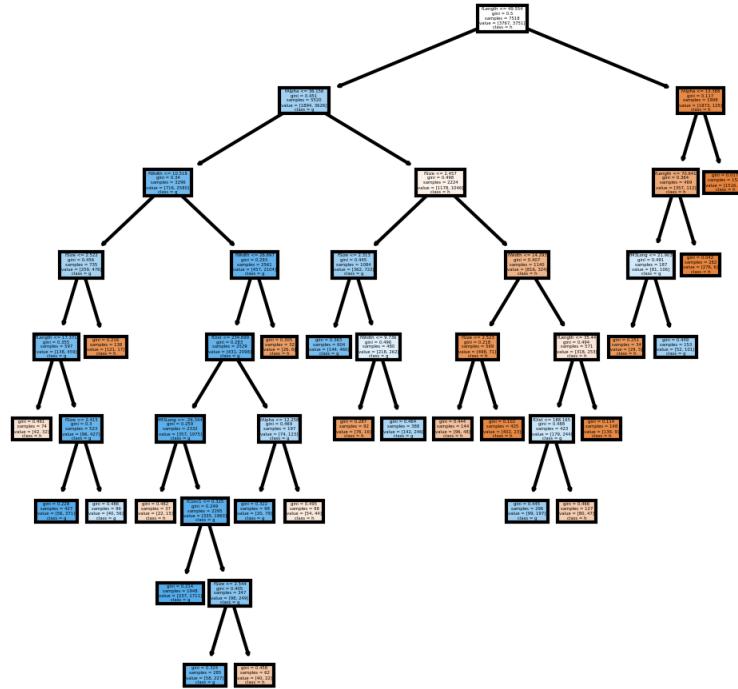
Nearmiss:



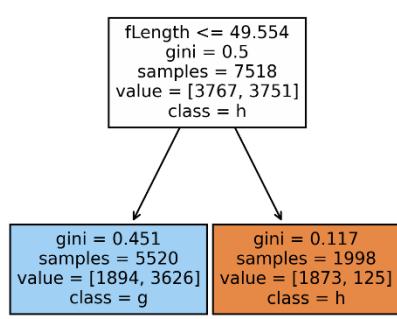
Metriche di valutazione dTree

```
Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:  
    Tuned  
Accuracy      0.84  
Error rate    0.16  
TP            1155.00  
TPR           0.92  
TNR           0.76  
FPR           0.24  
FNR           0.08  
Precision     0.80  
Recall        0.92  
F1            0.85  
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:  
          0.1000  0.0500  0.0300  0.0080  0.0075  0.0050  
Accuracy      0.71    0.71    0.74    0.78    0.78    0.79  
Error rate    0.29    0.29    0.26    0.22    0.22    0.21  
TP            1211.00 1211.00 859.00 1109.00 1183.00 1227.00  
TPR           0.96    0.96    0.68    0.88    0.94    0.97  
TNR           0.46    0.46    0.79    0.69    0.63    0.61  
FPR           0.54    0.54    0.21    0.31    0.37    0.39  
FNR           0.04    0.04    0.32    0.12    0.06    0.03  
Precision     0.64    0.64    0.77    0.74    0.72    0.72  
Recall        0.96    0.96    0.68    0.88    0.94    0.97  
F1            0.77    0.77    0.72    0.80    0.81    0.82
```

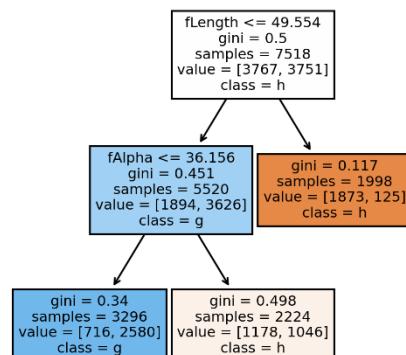
Albero tuned dataset processato



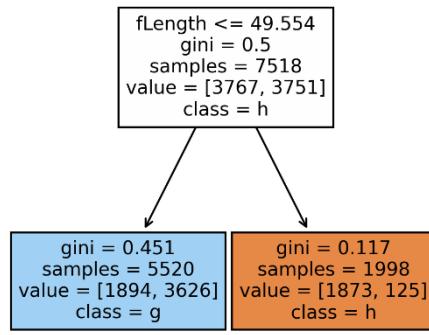
Tree pre-processato con mingain 0.1



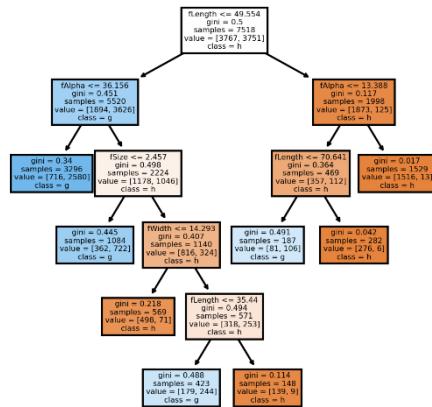
Tree pre-processato con mingain 0.03



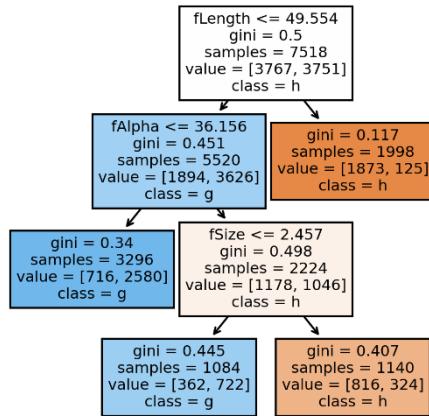
Tree pre-processato con mingain 0.05



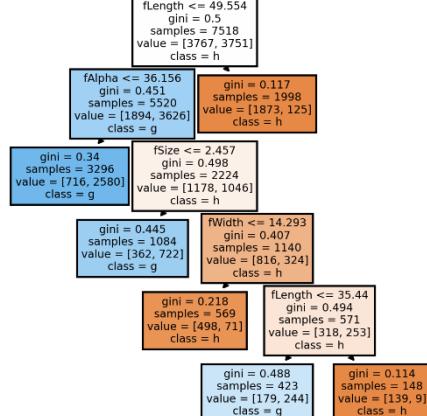
Tree pre-processato con mingain 0.005



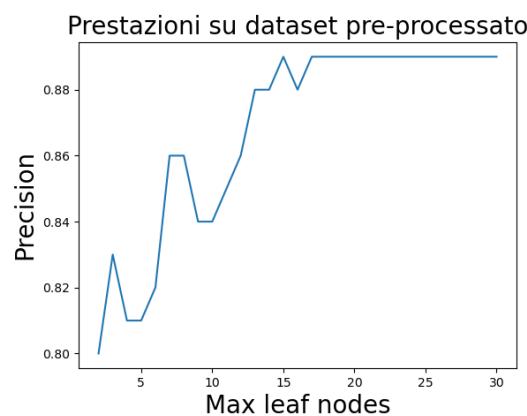
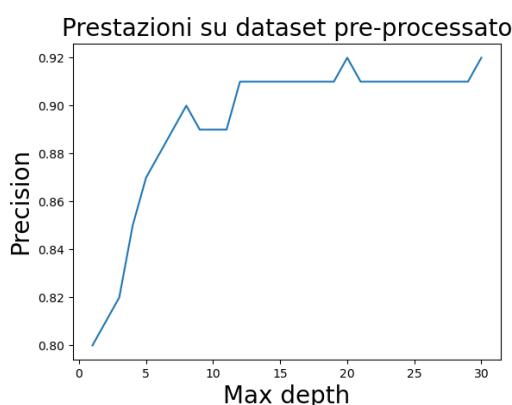
Tree pre-processato con mingain 0.008



Tree pre-processato con mingain 0.0075



Probabilistico:



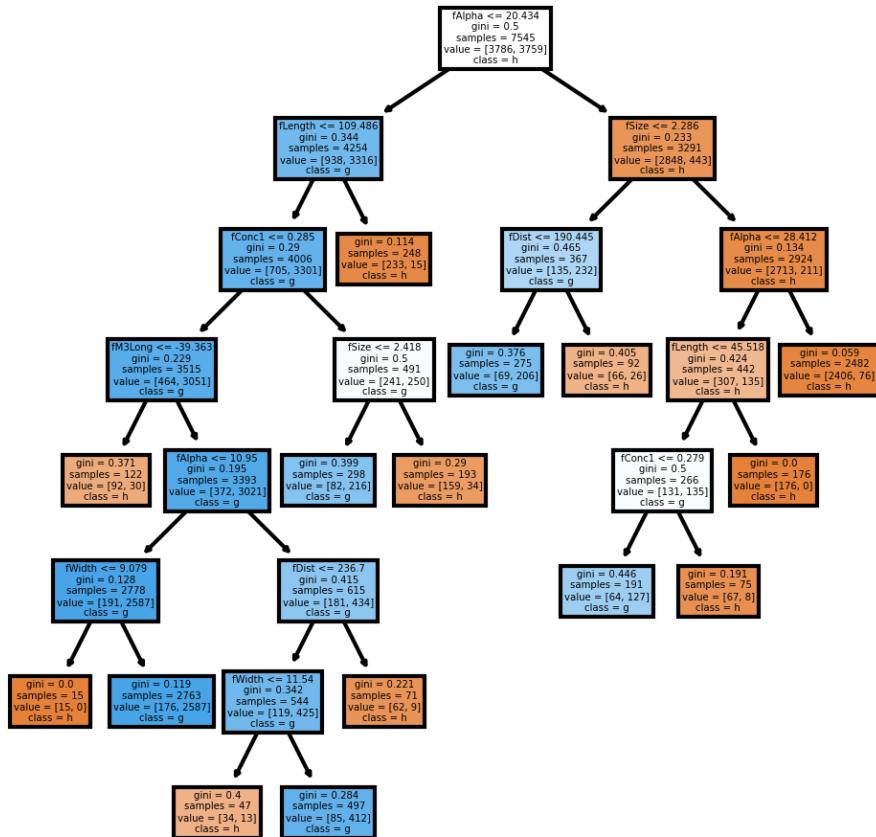
Metriche di valutazione dTree

```

Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:
          Tuned
Accuracy      0.91
Error rate    0.09
TP            1210.00
TPR           0.94
TNR           0.87
FPR           0.13
FNR           0.06
Precision     0.89
Recall        0.94
F1            0.91
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:
          0.1000  0.0500  0.0300  0.0080  0.0075  0.0050
Accuracy      0.83    0.83    0.85    0.88    0.88    0.89
Error rate    0.17    0.17    0.15    0.12    0.12    0.11
TP            1152.00 1152.00 1152.00 1192.00 1192.00 1228.00
TPR           0.89    0.89    0.89    0.92    0.92    0.95
TNR           0.76    0.76    0.81    0.84    0.84    0.82
FPR           0.24    0.24    0.19    0.16    0.16    0.18
FNR           0.11    0.11    0.11    0.08    0.08    0.05
Precision     0.80    0.80    0.83    0.86    0.86    0.85
Recall        0.89    0.89    0.89    0.92    0.92    0.95
F1            0.84    0.84    0.86    0.89    0.89    0.90

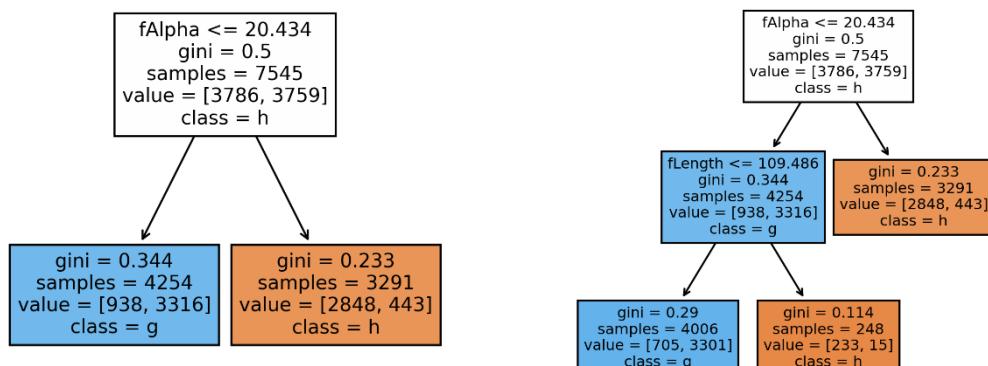
```

Albero tuned dataset processato

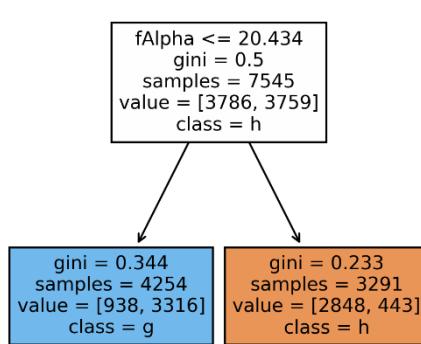


Tree pre-processato con mingain 0.1

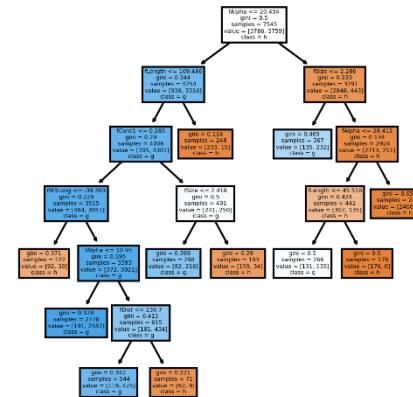
Tree pre-processato con mingain 0.03



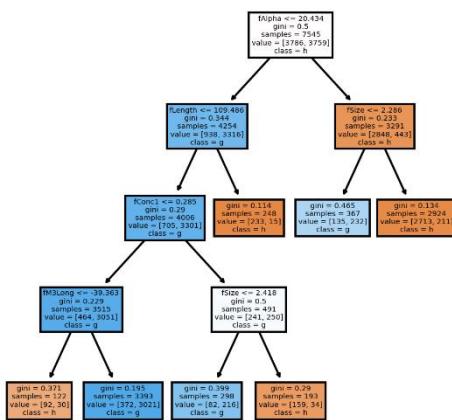
Tree pre-processato con mingain 0.05



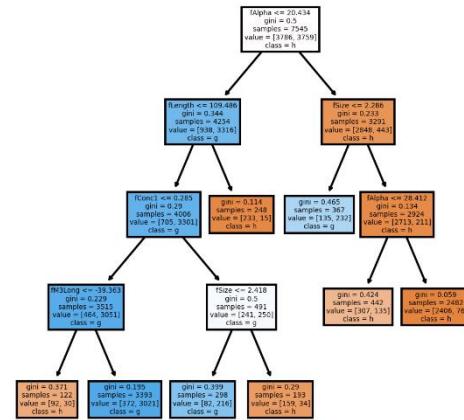
Tree pre-processato con mingain 0.005



Tree pre-processato con mingain 0.008



Tree pre-processato con mingain 0.0075



Utilizzando gli algoritmi di undersampling NearMiss e probabilistico l'albero tuned risulta di una complessità paragonabile o maggiore rispetto a quello ottenuto sui dati grezzi. Nel caso del probabilistico si riesce a raggiungere una precisione di 0.89. Il min gain 0.1 è in grado di ottenere uno split, cosa che non accadeva utilizzando i dati grezzi. Per quanto riguarda l'undersampling con random l'albero tuned è invece notevolmente più semplice rispetto a quello indotto con i dati grezzi, con una precisione di 0.80 ma classificando solo 796 TP.

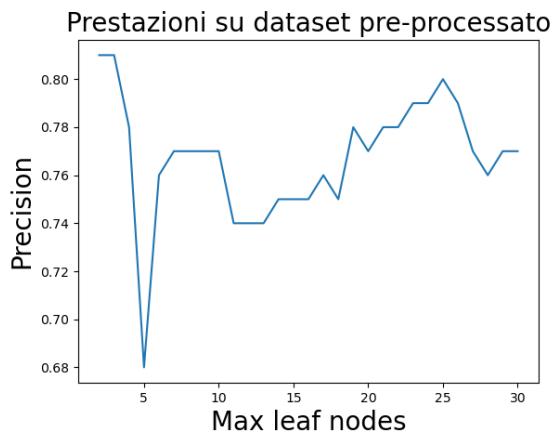
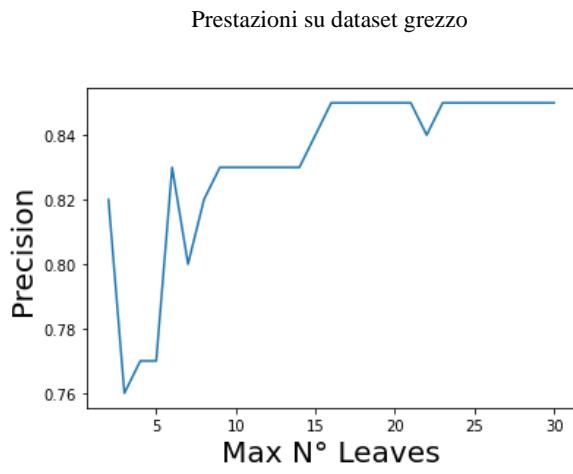
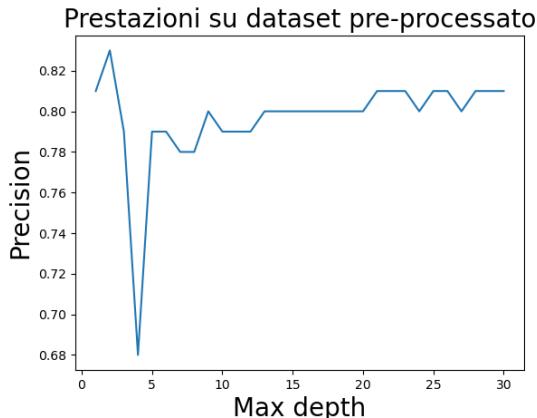
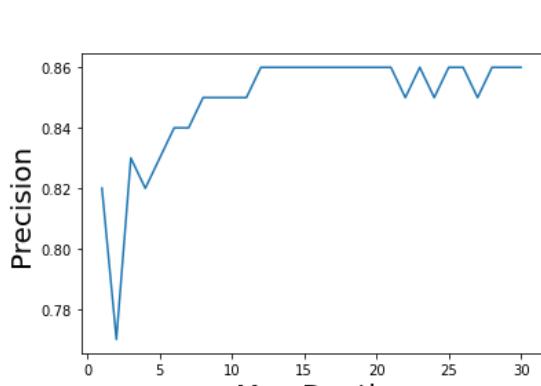
In generale le prestazioni con l'undersampling non sono molto migliori di quelle ottenute con i dati grezzi, ad eccezione per l'approccio probabilistico. Con esso le prestazioni di tutti gli alberi aumentano, ma non riescono a raggiungere la precisione di 0.97 ottenuta con il nearest neighbor.

Oversampling

Al contrario del kNN gli alberi decisionali hanno migliori prestazioni utilizzando l'undersampling piuttosto che l'oversampling. Nessuno degli alberi indotti riesce a superare di valori significativi le prestazioni ottenute con i dati grezzi e anzi esse sono spesso minori.

ADASYN:

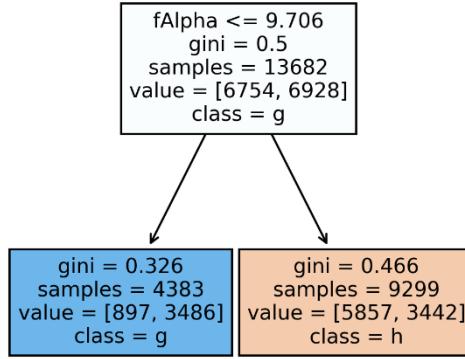
Prestazioni su dataset grezzo



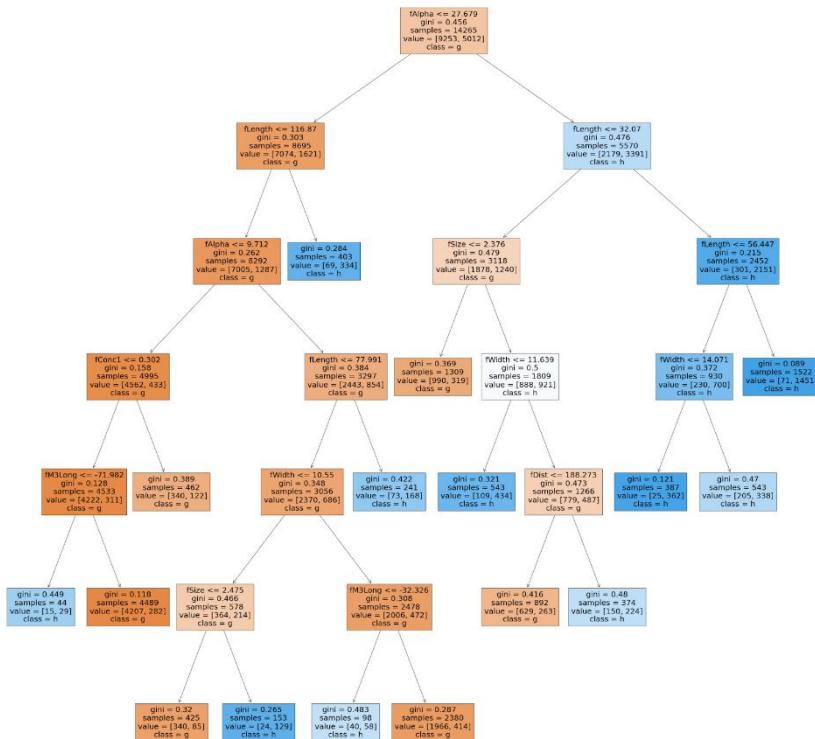
Metriche di valutazione dTree						
Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:						
Tuned						
Accuracy	0.68					
Error rate	0.32					
TP	1133.00					
TPR	0.49					
TNR	0.88					
FPR	0.12					
FNR	0.51					
Precision	0.81					
Recall	0.49					
F1	0.61					
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:						
	0.1000	0.0500	0.0300	0.0080	0.0075	0.0050
Accuracy	0.51	0.68	0.68	0.75	0.75	0.76
Error rate	0.49	0.32	0.32	0.25	0.25	0.24
TP	2325.00	1133.00	1133.00	1759.00	1759.00	1740.00
TPR	1.00	0.49	0.49	0.76	0.76	0.75
TNR	0.00	0.88	0.88	0.75	0.75	0.76
FPR	1.00	0.12	0.12	0.25	0.25	0.24
FNR	0.00	0.51	0.51	0.24	0.24	0.25
Precision	0.51	0.81	0.81	0.76	0.76	0.77
Recall	1.00	0.49	0.49	0.76	0.76	0.75
F1	0.68	0.61	0.61	0.76	0.76	0.76

	Tuned	0.1000	0.0500	0.0300	0.0080	0.0075	0.0050
Accuracy	0.83	Accuracy	0.65	0.74	0.78	0.79	0.82
Error rate	0.17	Error rate	0.35	0.26	0.22	0.21	0.18
TP	2796.00	TP	3079.00	2392.00	2984.00	2963.00	2930.00
TPR	0.91	TPR	1.00	0.78	0.97	0.96	0.95
TNR	0.70	TNR	0.00	0.68	0.43	0.48	0.57
FPR	0.30	FPR	1.00	0.32	0.57	0.52	0.43
FNR	0.09	FNR	0.00	0.22	0.03	0.04	0.05
Precision	0.85	Precision	0.65	0.82	0.76	0.77	0.80
Recall	0.91	Recall	1.00	0.78	0.97	0.96	0.95
F1	0.88	F1	0.79	0.80	0.85	0.86	0.87

Albero tuned dataset processato



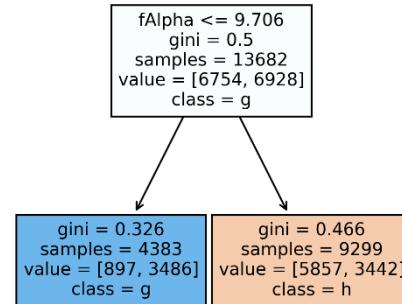
Albero tuned dataset grezzo



Tree pre-processato con mingain 0.1

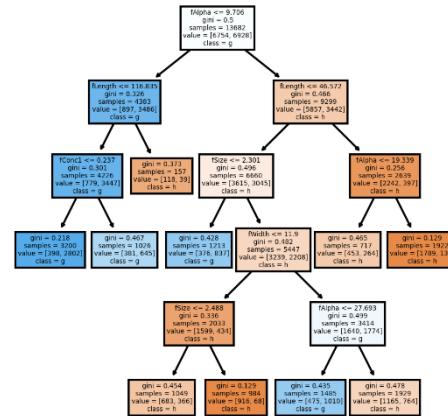
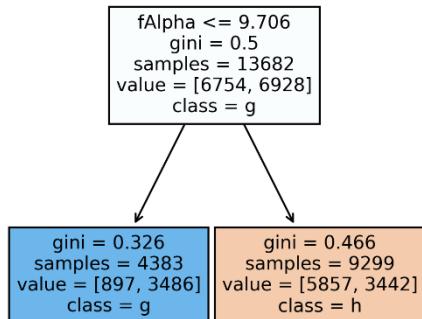
```
gini = 0.5
samples = 13682
value = [6754, 6928]
class = g
```

Tree pre-processato con mingain 0.03



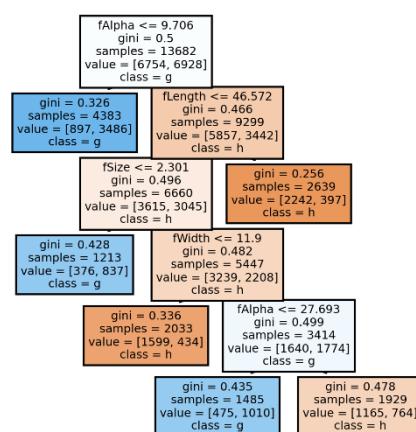
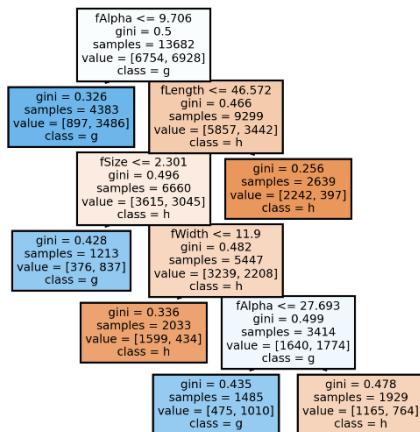
Tree pre-processato con mingain 0.05

Tree pre-processato con mingain 0.005

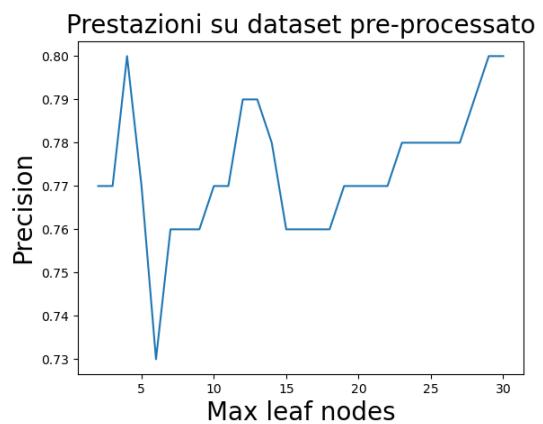
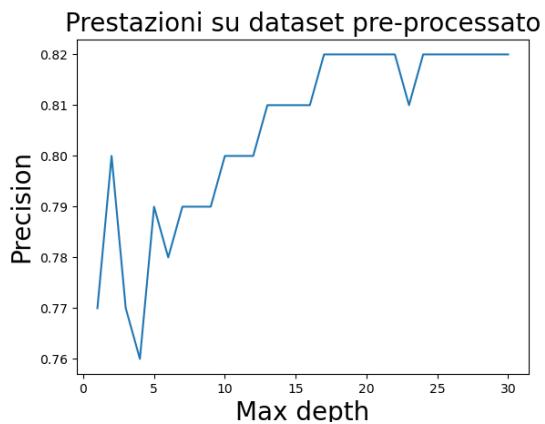


Tree pre-processato con mingain 0.008

Tree pre-processato con mingain 0.0075



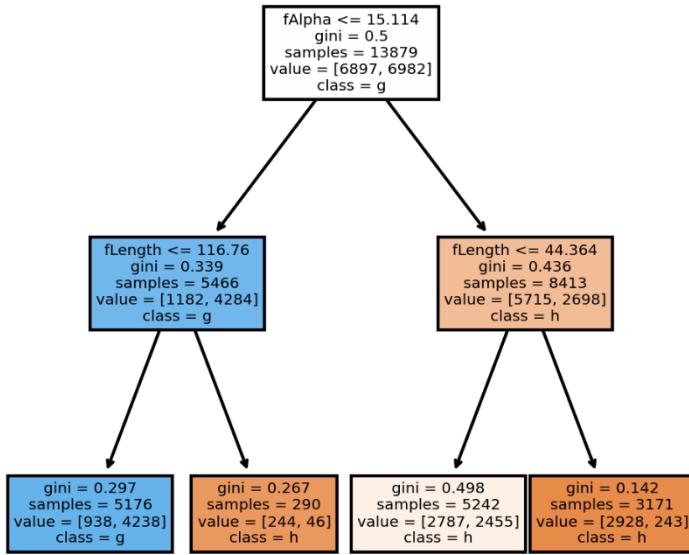
SMOTE:



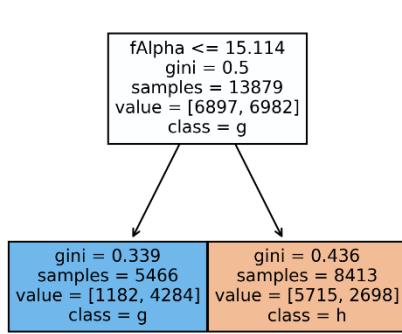
Metriche di valutazione dTree

```
Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:  
Tuned  
Accuracy      0.73  
Error rate    0.27  
TP            1375.00  
TPR           0.61  
TNR           0.85  
FPR           0.15  
FNR           0.39  
Precision     0.80  
Recall        0.61  
F1            0.69  
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:  
0.1000  0.0500  0.0300  0.0080  0.0075  0.0050  
Accuracy      0.72  0.72   0.72   0.78   0.78   0.79  
Error rate    0.28  0.28   0.28   0.22   0.22   0.21  
TP            1393.00 1393.00 1393.00 1946.00 1946.00 1871.00  
TPR           0.61  0.61   0.61   0.86   0.86   0.82  
TNR           0.82  0.82   0.82   0.70   0.70   0.77  
FPR           0.18  0.18   0.18   0.30   0.30   0.23  
FNR           0.39  0.39   0.39   0.14   0.14   0.18  
Precision     0.77  0.77   0.77   0.73   0.73   0.77  
Recall        0.61  0.61   0.61   0.86   0.86   0.82  
F1            0.68  0.68   0.68   0.79   0.79   0.80
```

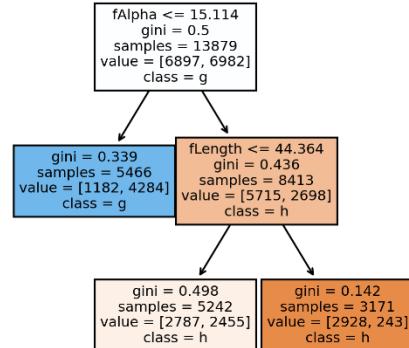
Albero tuned dataset processato



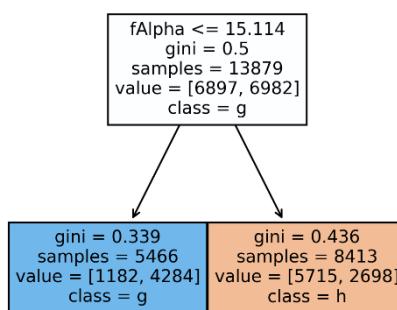
Tree pre-processato con mingain 0.1



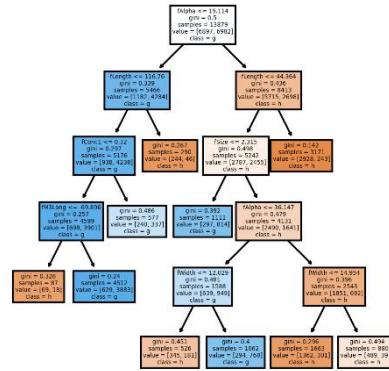
Tree pre-processato con mingain 0.03



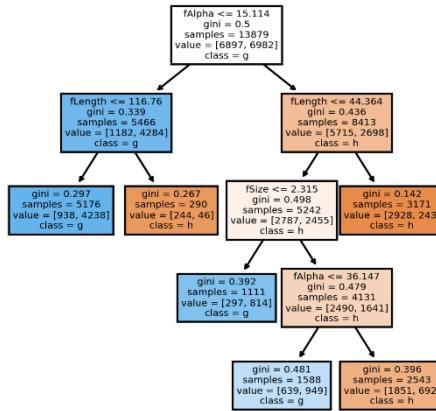
Tree pre-processato con mingain 0.05



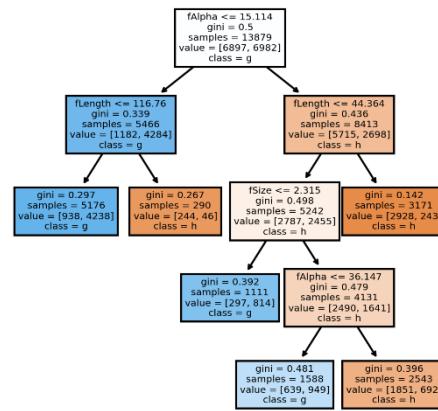
Tree pre-processato con mingain 0.005



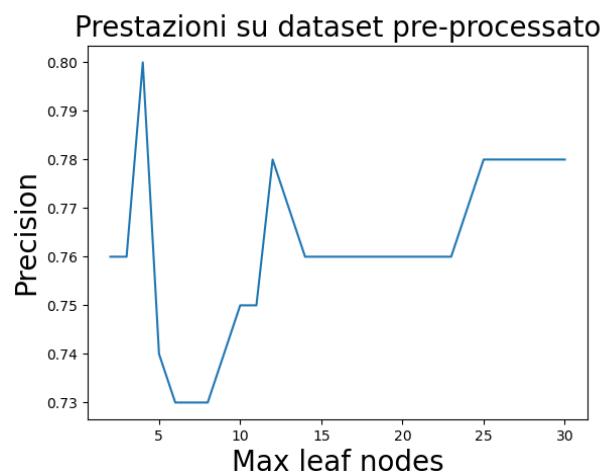
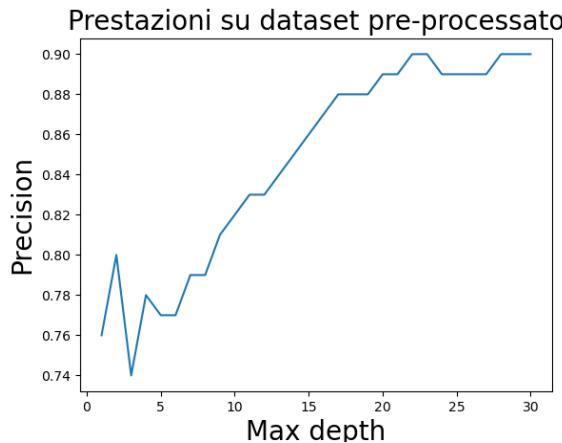
Tree pre-processato con mingain 0.008



Tree pre-processato con mingain 0.0075



RandomOverSampler:



Metriche di valutazione dTree

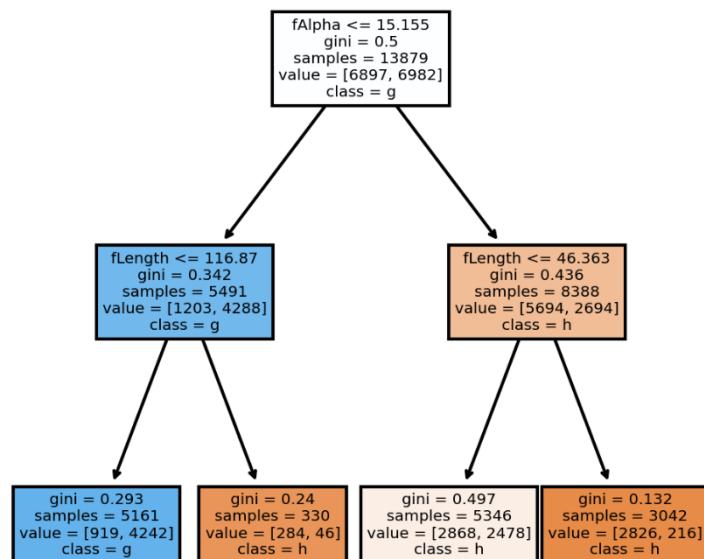
```

Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:
    Tuned
Accuracy      0.73
Error rate    0.27
TP            1379.00
TPR           0.61
TNR           0.86
FPR           0.14
FNR           0.39
Precision     0.80
Recall        0.61
F1            0.69

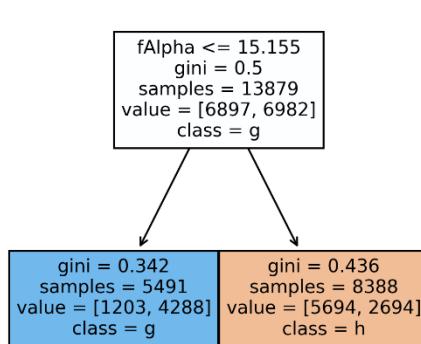
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:
          0.1000  0.0500  0.0300  0.0080  0.0075  0.0050
Accuracy      0.72   0.72   0.72   0.78   0.78   0.78
Error rate    0.28   0.28   0.28   0.22   0.22   0.22
TP            1397.00 1397.00 1397.00 1959.00 1959.00 1959.00
TPR           0.62   0.62   0.62   0.86   0.86   0.86
TNR           0.82   0.82   0.82   0.70   0.70   0.70
FPR           0.18   0.18   0.18   0.30   0.30   0.30
FNR           0.38   0.38   0.38   0.14   0.14   0.14
Precision     0.76   0.76   0.76   0.73   0.73   0.73
Recall        0.62   0.62   0.62   0.86   0.86   0.86
F1            0.68   0.68   0.68   0.79   0.79   0.79

```

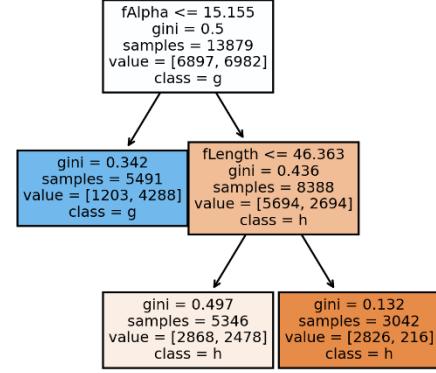
Albero tuned dataset processato



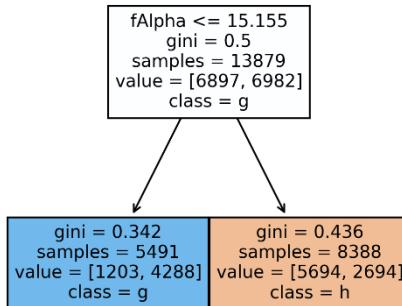
Tree pre-processato con mingain 0.1



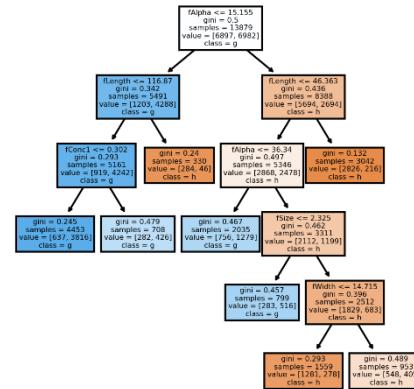
Tree pre-processato con mingain 0.03



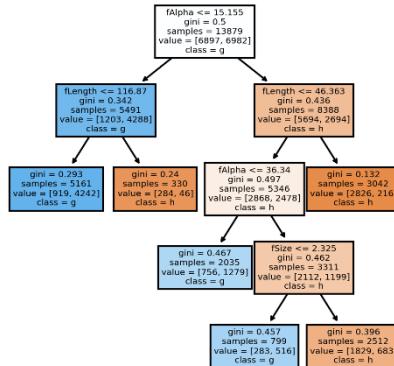
Tree pre-processato con mingain 0.05



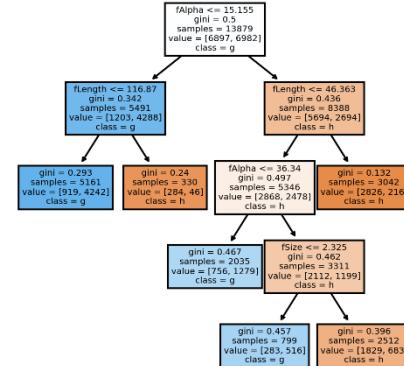
Tree pre-processato con mingain 0.005



Tree pre-processato con mingain 0.008



Tree pre-processato con mingain 0.0075



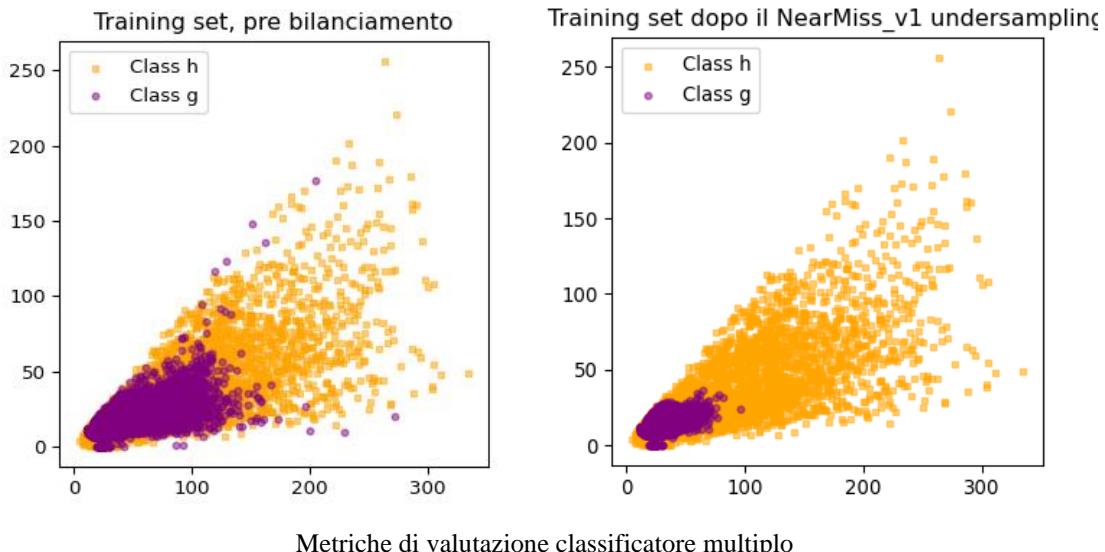
Classificatore custom: Classificatore multiplo

Undersampling

Nearmiss:

Il nearmiss evidenza un deterioramento della precisione significativo. In particolare, il classificatore Clf_3, subisce fortemente e in modo negativo questa tecnica di undersampling, passando da una precisione di 0.82 a una precisione del 0.64.

E' interessante analizzare questo tipo di tecnica comparandola alle prestazioni ottenute, dato che è visivamente notabile meno rumore per quanto riguarda la classe g, e quindi una distinzione graficamente netta. Questo vuol dire che molto probabilmente la riduzione applicata al dataset è stata troppo significativa, tale da avere un impatto negativo sulle capacità del modello.



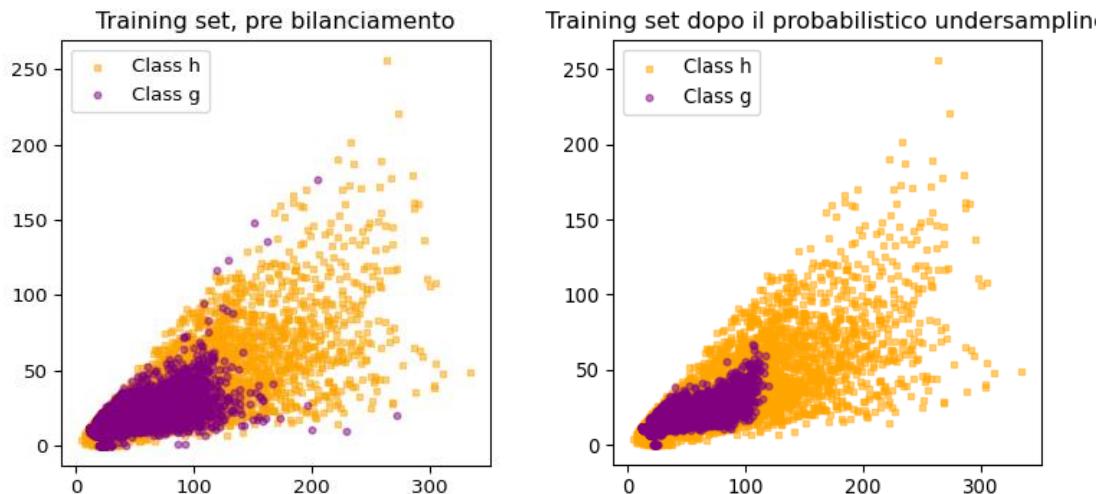
Metriche di valutazione classificatore multiplo

Eseguo classificatore multiplo custom su dataset grezzo...				Metriche di valutazione, test set, su dataset pre-processato:				
	Clf_1	Clf_2	Clf_3	Clf_final	Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.83	0.82	0.74	0.83	0.83	0.79	0.71	0.79
Error rate	0.17	0.18	0.26	0.17	0.17	0.21	0.29	0.21
TP	2796.00	2885.00	2392.00	2831.00	1197.00	1227.00	1211.00	1227.00
TPR	0.91	0.94	0.78	0.92	0.95	0.97	0.96	0.97
TNR	0.70	0.61	0.68	0.66	0.70	0.61	0.46	0.61
FPR	0.30	0.39	0.32	0.34	0.30	0.39	0.54	0.39
FNR	0.09	0.06	0.22	0.08	0.05	0.03	0.04	0.03
Precision	0.85	0.82	0.82	0.83	0.76	0.72	0.64	0.72
Recall	0.91	0.94	0.78	0.92	0.95	0.97	0.96	0.97
F1	0.88	0.87	0.80	0.87	0.85	0.82	0.77	0.82

Probabilistico:

La strategia di undersampling probabilistico è emersa come un valido miglioramento delle prestazioni per la maggior parte dei classificatori provati. In questo specifico caso si nota una somiglianza notevole con Clf_1, ma con un miglioramento leggermente superiore e quasi addirittura irrilevante.

Il classificatore complessivo raggiunge una precisione del 0.88 e un'accuratezza del 0.91 con un numero di TP pari a 1227.



Metriche di valutazione classificatore multiplo

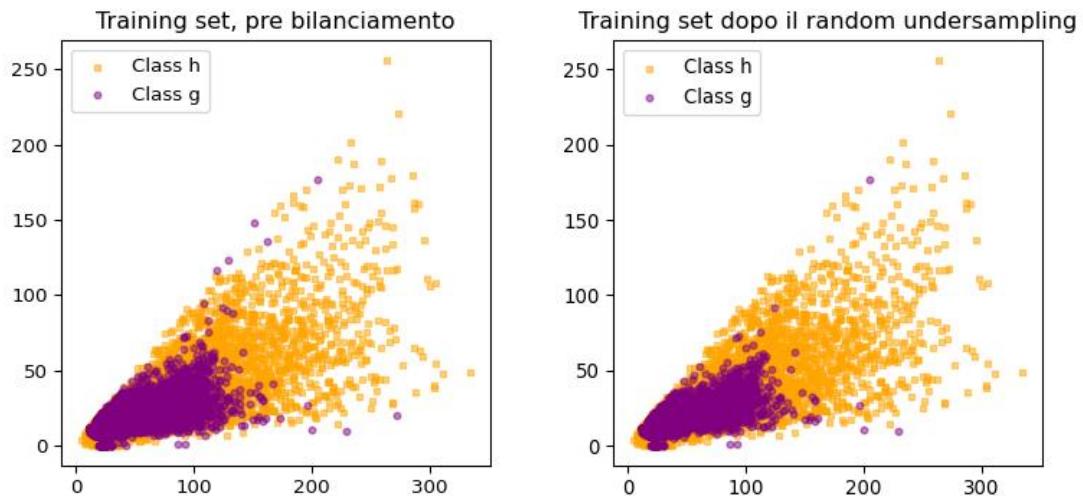
Metriche di valutazione, test set, su dataset pre-processato:

	Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.91	0.89	0.83	0.91
Error rate	0.09	0.11	0.17	0.09
TP	1223.00	1228.00	1152.00	1227.00
TPR	0.95	0.95	0.89	0.95
TNR	0.87	0.82	0.76	0.86
FPR	0.13	0.18	0.24	0.14
FNR	0.05	0.05	0.11	0.05
Precision	0.88	0.85	0.80	0.88
Recall	0.95	0.95	0.89	0.95
F1	0.91	0.90	0.84	0.91

Random UnderSampling:

Il random undersampling potrebbe non essere la scelta ottimale, ma nel caso specifico dimostra comunque delle prestazioni migliori rispetto alla tecnica di NearMiss_v1. Nonostante questa scelta, non è facilmente notabile attraverso lo scatter plot una netta distinzione tra dataset originale e pre-processato.

Questo suggerisce che, le limitazioni del random oversampling non riescono a contribuire a un miglioramento del modello.



Metriche di valutazione classificatore multiplo

	Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.79	0.79	0.72	0.79
Error rate	0.21	0.21	0.28	0.21
TP	1053.00	1070.00	807.00	1070.00
TPR	0.84	0.85	0.64	0.85
TNR	0.75	0.72	0.81	0.73
FPR	0.25	0.28	0.19	0.27
FNR	0.16	0.15	0.36	0.15
Precision	0.77	0.76	0.77	0.76
Recall	0.84	0.85	0.64	0.85
F1	0.80	0.80	0.70	0.80

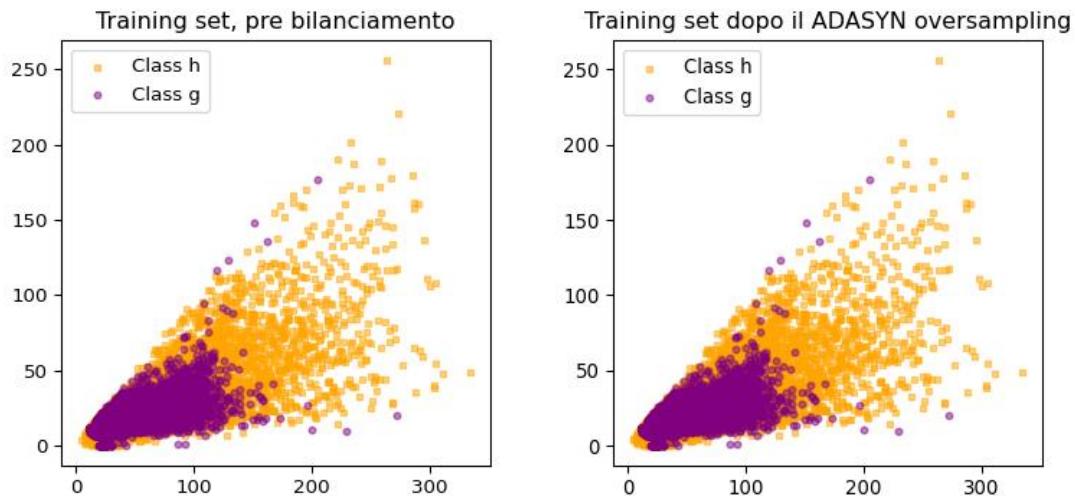
Oversampling

ADASYN:

Possiamo notare un impatto “marginale” sulla precisione del Clf_3, ma sicuramente notevole sui restanti, compreso di conseguenza il classificatore finale. In termini pratici, i dati ottenuti ci suggeriscono che non si tratta di una tecnica di oversampling idonea al classificatore vista la sua troppa influenza negativa.

Possiamo notare come in realtà l’applicazione di una tecnica di oversampling (in questo caso abbiamo analizzato la tecnica ADASYN ma il discorso è comunque più generale) su una classe fortemente minoritaria come quella h, può comportare una distribuzione aumentata e quindi di conseguenza delle peggiori prestazioni.

In questo caso noi non abbiamo fatto altro che rendere il lavoro più complicato al modello.

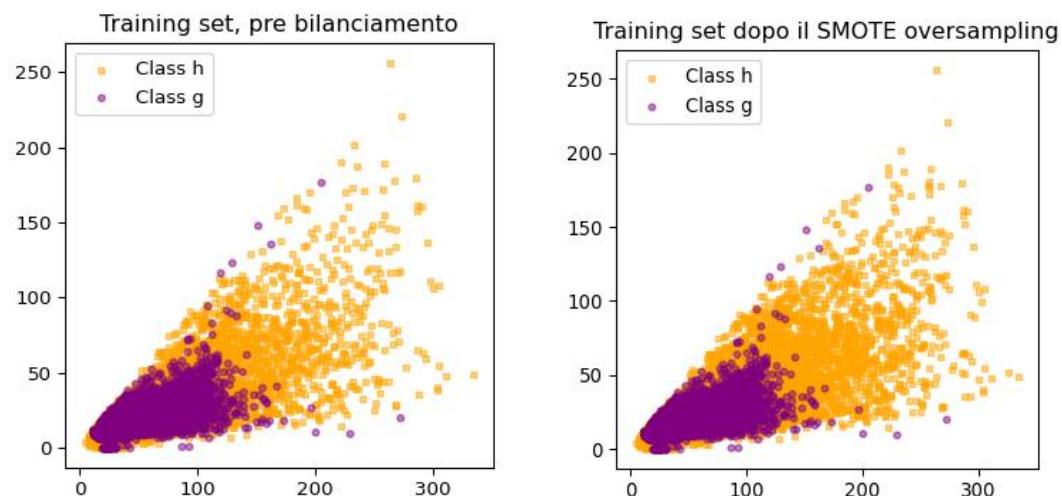


Metriche di valutazione classificatore multiplo

Eseguo classificatore multiplo custom su dataset grezzo...				Metriche di valutazione, test set, su dataset pre-processato:			
Metriche di valutazione, test set, su dataset grezzo:				Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.83	0.82	0.74	0.83	0.75	0.76	0.68
Error rate	0.17	0.18	0.26	0.17	0.25	0.24	0.32
TP	2796.00	2885.00	2392.00	2831.00	1819.00	1740.00	1133.00
TPR	0.91	0.94	0.78	0.92	0.78	0.75	0.49
TNR	0.70	0.61	0.68	0.66	0.72	0.76	0.88
FPR	0.30	0.39	0.32	0.34	0.28	0.24	0.12
FNR	0.09	0.06	0.22	0.08	0.22	0.25	0.51
Precision	0.85	0.82	0.82	0.83	0.75	0.77	0.77
Recall	0.91	0.94	0.78	0.92	0.78	0.75	0.49
F1	0.88	0.87	0.80	0.87	0.76	0.76	0.61

SMOTE:

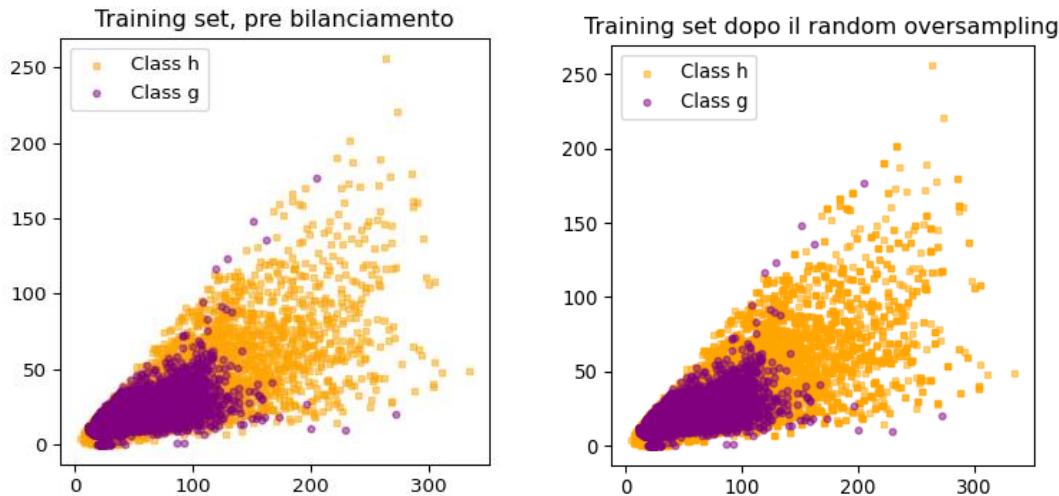
Le affermazioni appena citate per quanto riguarda la tecnica ADASYN vengono confermate tramite l'utilizzo della tecnica di oversampling SMOTE e random, che mostrano comunque un deterioramento significativo delle prestazioni.



Metriche di valutazione classificatore multiplo				
Metriche di valutazione, test set, su dataset pre-processato:				
	Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.81	0.79	0.72	0.79
Error rate	0.19	0.21	0.28	0.21
TP	2000.00	1871.00	1393.00	1871.00
TPR	0.88	0.82	0.61	0.82
TNR	0.74	0.77	0.82	0.77
FPR	0.26	0.23	0.18	0.23
FNR	0.12	0.18	0.39	0.18
Precision	0.76	0.77	0.77	0.77
Recall	0.88	0.82	0.61	0.82
F1	0.82	0.80	0.68	0.80

RandomOverSampler:

Notiamo nuovamente un deterioramento delle prestazioni, che conferma ciò detto nel paragrafo relativo alla tecnica ADASYN. In questo caso il deterioramento è il più significativo per quanto riguarda questo modello.



Metriche di valutazione classificatore multiplo				
Metriche di valutazione, test set, su dataset pre-processato:				
	Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.80	0.78	0.72	0.79
Error rate	0.20	0.22	0.28	0.21
TP	1985.00	1959.00	1397.00	1949.00
TPR	0.87	0.86	0.62	0.86
TNR	0.73	0.70	0.82	0.73
FPR	0.27	0.30	0.18	0.27
FNR	0.13	0.14	0.38	0.14
Precision	0.76	0.73	0.76	0.75
Recall	0.87	0.86	0.62	0.86
F1	0.81	0.79	0.68	0.80

4.2. Confronto features

Ci sono delle osservazioni generali che conviene riportare prima di mostrare i risultati. In tutti i casi nei quali si è deciso di usare i pesi nelle distanze del k-NN il classificatore è andato incontro ad un problema di overfitting, problema che si è verificato anche nel bilanciamento con undersampling e oversampling mostrato precedentemente. Sempre in questo tipo di classificatore non si è percepita una variazione notevole dei risultati tra distanza euclidea e distanza Manhattan, portando nella fase successiva del tuning a scegliere la seconda per un minor costo computazionale. Nelle immagini che indicano il miglior tuning la distanza è indicata dal valore “ p ”, e se $p = 1$ si tratta di distanza Manhattan, se $p=2$ si tratta di distanza euclidea.

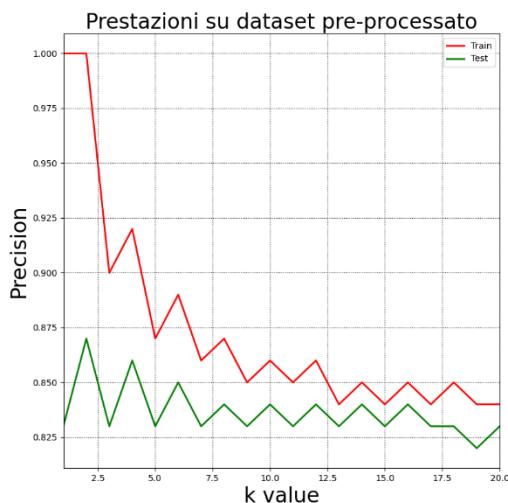
Confronto standardizzazione dei dati

Come detto precedentemente nella standardizzazione dei dati si è optato per la standardizzazione MinMax. Di seguito si analizzano i dati sottoposti a questa strategia per i tre classificatori.

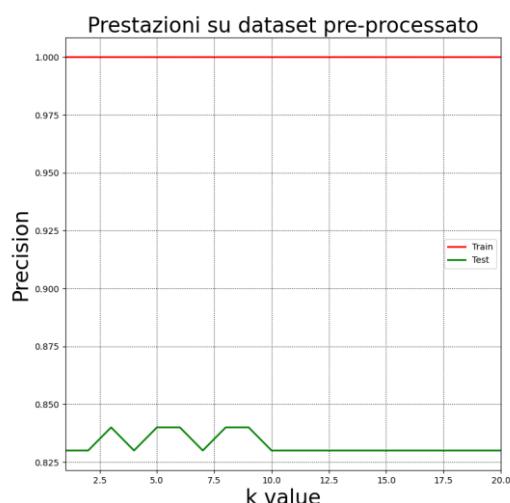
KNN

Dai grafici si può notare come si crei un notevole problema di overfitting (con conseguente calo delle prestazioni sul test set) pesando le distanze, mentre usando la distanza euclidea o quella Manhattan la precisione rimane quasi invariata.

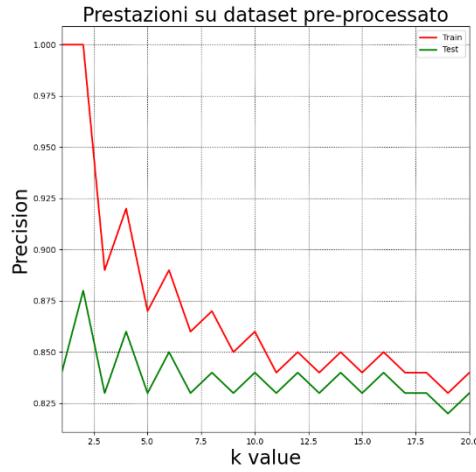
Prestazioni kNN con diversi k value



Prestazioni kNN con diversi k value con distanze pesate

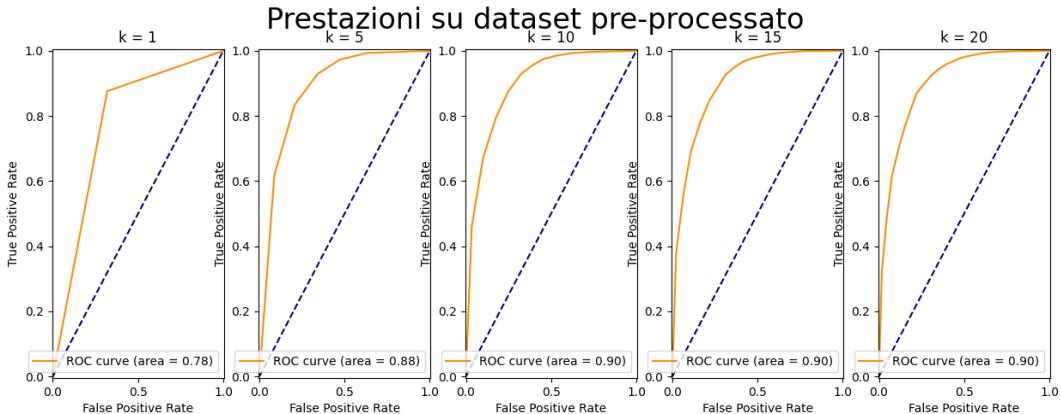


Prestazioni kNN con diversi k value con distanza di manhattan

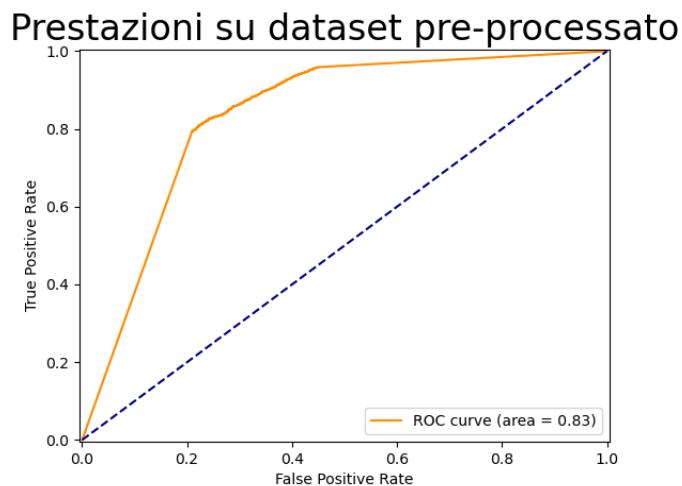


Per la creazione della curva ROC si nota come bastino 5 vicini per raggiungere il massimo delle prestazioni del modello, con l'immagine in alto che si riferisce sia alla distanza euclidea che alla distanza Manhattan, oltre che al valore trovato pesando le feature. Dopo il tuning degli iperparametri si è scelto un metodo che potesse massimizzare le prestazioni con il costo computazionale minimo, ottenendo la curva ROC in basso.

Prestazioni kNN distanza euclidea, distanza Manhattan e distanze pesate



Prestazioni kNN tuned



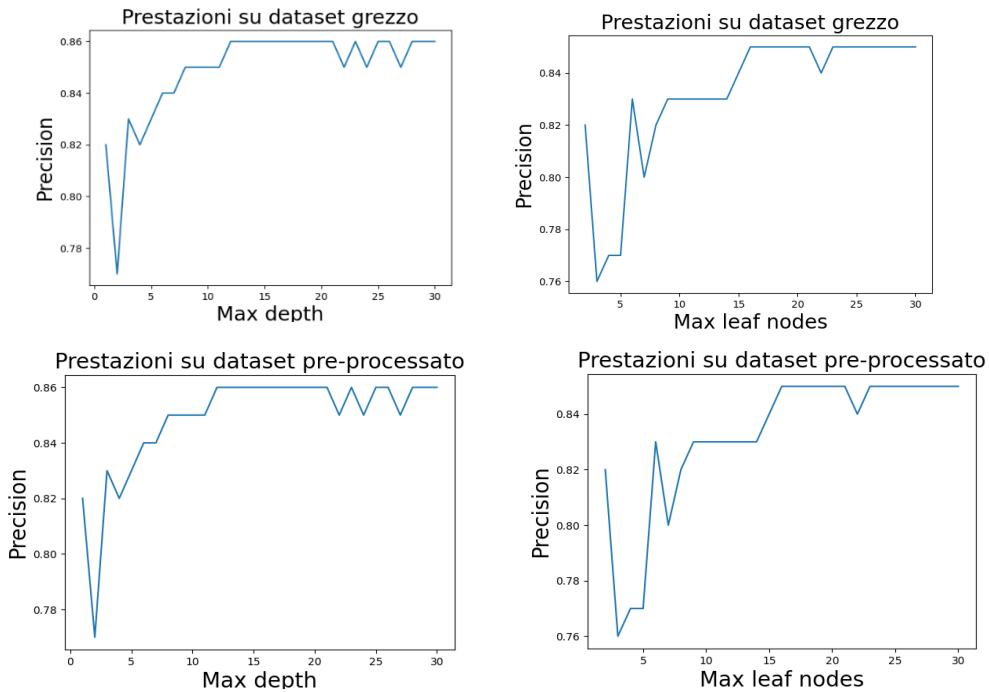
Di seguito sono riportate le tabelle con le metriche di valutazione per effettuare il tuning degli iperparametri (le prime tre tabelle) con le tre strategie prima definite, mentre la quarta tabella indica i valori scelti per il modello tuned.

Metriche di valutazione kNN

Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.81	0.79	0.83	0.83	0.83	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	
Error rate	0.19	0.21	0.17	0.17	0.17	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	
TP	2698.00	2430.00	2815.00	2694.00	2863.00	2786.00	2887.00	2829.00	2905.00	2861.00	2909.00	2877.00	2914.00	2885.00	2925.00	2896.00	2927.00	2908.00	2931.00	2907.00	
TPR	0.88	0.79	0.91	0.87	0.93	0.90	0.94	0.92	0.94	0.93	0.94	0.93	0.95	0.94	0.95	0.94	0.95	0.94	0.95	0.94	
TNR	0.68	0.79	0.66	0.74	0.66	0.71	0.65	0.68	0.65	0.68	0.64	0.67	0.64	0.66	0.63	0.66	0.63	0.65	0.63	0.65	
FPR	0.32	0.21	0.34	0.26	0.34	0.29	0.35	0.32	0.35	0.32	0.36	0.33	0.34	0.37	0.34	0.37	0.35	0.37	0.35	0.37	
FNR	0.12	0.21	0.09	0.13	0.07	0.10	0.06	0.08	0.06	0.07	0.06	0.07	0.05	0.06	0.05	0.06	0.05	0.06	0.05	0.06	
Precision	0.83	0.87	0.83	0.86	0.83	0.85	0.83	0.84	0.83	0.84	0.83	0.84	0.83	0.84	0.83	0.84	0.83	0.83	0.82	0.83	
Recall	0.88	0.79	0.91	0.87	0.93	0.90	0.94	0.92	0.94	0.93	0.94	0.93	0.95	0.94	0.95	0.94	0.95	0.94	0.95	0.94	
F1	0.85	0.83	0.87	0.87	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	
il valore di k che porta precisione migliore è 2 che classifica 2430.0 TP con precisione 0.87 senza cambiare alcun iperparametro eccetto k																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.81	0.81	0.83	0.83	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.85	0.84	0.85	
Error rate	0.19	0.19	0.17	0.17	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.15	0.16	0.15	
TP	2698.00	2698.00	2812.00	2817.00	2860.00	2866.00	2887.00	2895.00	2900.00	2906.00	2906.00	2913.00	2914.00	2920.00	2924.00	2923.00	2935.00	2930.00	2935.00	2935.00	
TPR	0.88	0.88	0.91	0.91	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	
TNR	0.68	0.68	0.67	0.66	0.67	0.67	0.66	0.66	0.66	0.66	0.66	0.65	0.65	0.65	0.65	0.65	0.65	0.64	0.65	0.65	
FPR	0.32	0.32	0.33	0.34	0.33	0.33	0.34	0.34	0.34	0.34	0.34	0.35	0.35	0.35	0.35	0.35	0.35	0.36	0.35	0.36	
FNR	0.12	0.12	0.09	0.09	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	
Precision	0.83	0.83	0.84	0.83	0.84	0.84	0.83	0.84	0.84	0.83	0.84	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	
Recall	0.88	0.88	0.91	0.91	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	
F1	0.85	0.85	0.87	0.87	0.88	0.88	0.88	0.88	0.89	0.89	0.89	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	
il valore di k che porta precisione migliore è 3 che classifica 2812.0 TP con precisione 0.84 pesando le distanze																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.82	0.81	0.83	0.83	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.85	0.85	0.85	0.85	0.84	0.85	
Error rate	0.18	0.19	0.17	0.17	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.15	0.15	0.15	0.15	0.16	0.15	0.15	
TP	2748.00	2488.00	2853.00	2730.00	2894.00	2816.00	2911.00	2863.00	2929.00	2873.00	2935.00	2904.00	2942.00	2911.00	2948.00	2926.00	2954.00	2930.00	2954.00	2936.00	
TPR	0.89	0.81	0.93	0.89	0.94	0.91	0.95	0.93	0.95	0.93	0.95	0.94	0.96	0.95	0.96	0.95	0.96	0.95	0.96	0.95	
TNR	0.68	0.80	0.65	0.74	0.65	0.70	0.64	0.68	0.64	0.67	0.63	0.66	0.64	0.66	0.64	0.66	0.64	0.65	0.63	0.65	
FPR	0.32	0.20	0.35	0.26	0.35	0.30	0.36	0.32	0.36	0.33	0.37	0.34	0.37	0.34	0.36	0.35	0.37	0.35	0.37	0.35	
FNR	0.11	0.19	0.07	0.11	0.06	0.09	0.05	0.07	0.05	0.07	0.05	0.06	0.04	0.05	0.04	0.05	0.04	0.05	0.04	0.05	
Precision	0.84	0.88	0.83	0.86	0.83	0.85	0.83	0.84	0.83	0.84	0.83	0.84	0.83	0.84	0.83	0.84	0.83	0.83	0.82	0.83	
Recall	0.89	0.81	0.93	0.89	0.94	0.91	0.95	0.93	0.95	0.93	0.95	0.94	0.96	0.95	0.96	0.95	0.96	0.95	0.96	0.95	
F1	0.86	0.84	0.88	0.87	0.88	0.88	0.88	0.88	0.89	0.89	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	
il valore di k che porta precisione migliore è 2 che classifica 2488.0 TP con precisione 0.88 utilizzando la distanza Manhattan																					
Metriche di valutazione, test set:		2																			
Accuracy		0.81																			
Error rate		0.19																			
TP		2488.00																			
TPR		0.81																			
TNR		0.88																			
FPR		0.20																			
FNR		0.19																			
Precision		0.88																			
Recall		0.81																			
F1		0.84																			
il valore di k che porta precisione migliore è 2 che classifica 2488.0 TP con precisione 0.88 con k=2, weight=uniform e p=1																					

Dtree

Anche per la standardizzazione sono stati usati tre criteri per effettuare il tuning degli iperparametri: la massima profondità dell'albero, il massimo numero di foglie e il guadagno minimo degli split. Essendo il modello con l'albero decisionale meno dipendente dai differenti ordini di grandezza degli attributi rispetto al classificatore k-NN, non si notano delle differenze nei primi due criteri, come mostrato nelle immagini seguenti.



L'immagine che segue è invece relativa ai valori delle metriche dell'albero tuned (che utilizza come riferimento il numero di nodi e la profondità che massimizzano la precision) nella parte in alto. Le metriche relative ai vari tuning effettuati basandosi sul minimo guadagno sono invece nella parte inferiore dell'immagine.

Metriche di valutazione dTree

Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:						
	Tuned					
Accuracy	0.83					
Error rate	0.17					
TP	2796.00					
TPR	0.91					
TNR	0.70					
FPR	0.30					
FNR	0.09					
Precision	0.85					
Recall	0.91					
F1	0.88					
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:						
	0.1000	0.0500	0.0300	0.0080	0.0075	0.0050
Accuracy	0.65	0.74	0.78	0.79	0.82	0.82
Error rate	0.35	0.26	0.22	0.21	0.18	0.18
TP	3079.00	2392.00	2984.00	2963.00	2930.00	2885.00
TPR	1.00	0.78	0.97	0.96	0.95	0.94
TNR	0.00	0.68	0.43	0.48	0.57	0.61
FPR	1.00	0.32	0.57	0.52	0.43	0.39
FNR	0.00	0.22	0.03	0.04	0.05	0.06
Precision	0.65	0.82	0.76	0.77	0.80	0.82
Recall	1.00	0.78	0.97	0.96	0.95	0.94
F1	0.79	0.80	0.85	0.86	0.87	0.87

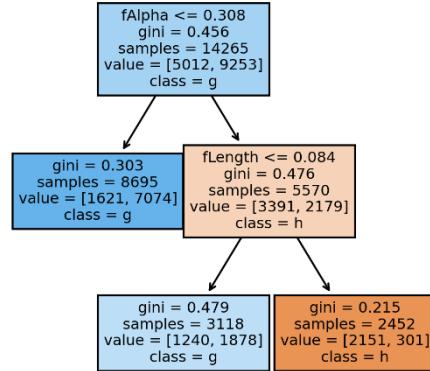
Tutte le immagini seguenti sono relative agli alberi creati seguendo le strategie elencate in precedenza; la complessità dell'albero tuned è elevata ma è accompagnata da una precisione notevole, mentre gli alberi trovati con il minimo guadagno hanno complessità inversamente proporzionale al valore scelto per il `min_gain`, poiché all'aumentare di questo valore si inasprisce il criterio di stop.



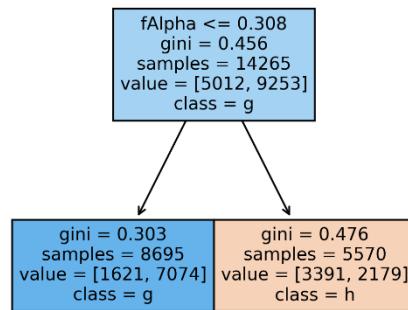
Tree pre-processato con `mingain` 0.1

**gini = 0.456
samples = 14265
value = [5012, 9253]
class = g**

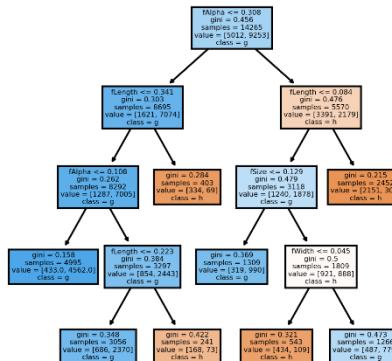
Tree pre-processato con `mingain` 0.03



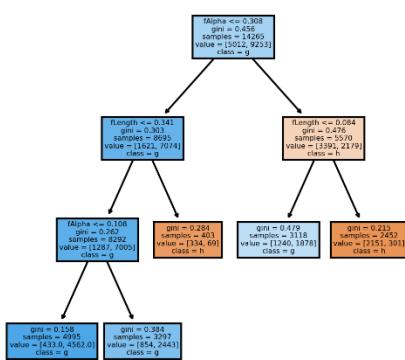
Tree pre-processato con mingain 0.05



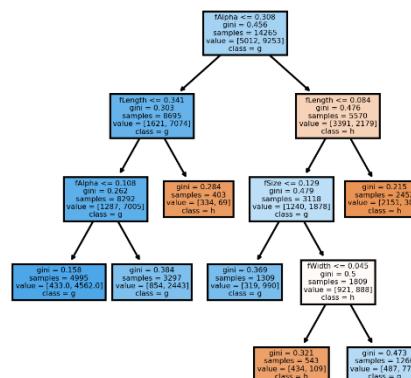
Tree pre-processato con mingain 0.005



Tree pre-processato con mingain 0.008



Tree pre-processato con mingain 0.0075



Classificatore multiplo

Anche nel classificatore multiplo non si nota differenza nelle prestazioni tra le metriche sul dataset grezzo o dataset processato. Questo è dovuto al fatto che i classificatori interni siano alberi decisionali e, come detto in precedenza, essi subiscono poco la differenza negli ordini di grandezza tra gli attributi.

Prestazioni sul dataset processato					Prestazioni sul dataset grezzo				
Metriche di valutazione, test set, su dataset pre-processato:					Eseguo classificatore multiplo custom su dataset grezzo...				
	Clf_1	Clf_2	Clf_3	Clf_final		Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.83	0.82	0.74	0.83	Accuracy	0.83	0.82	0.74	0.83
Error rate	0.17	0.18	0.26	0.17	Error rate	0.17	0.18	0.26	0.17
TP	2796.00	2885.00	2392.00	2831.00	TP	2796.00	2885.00	2392.00	2831.00
TPR	0.91	0.94	0.78	0.92	TPR	0.91	0.94	0.78	0.92
TNR	0.70	0.61	0.68	0.66	TNR	0.70	0.61	0.68	0.66
FPR	0.30	0.39	0.32	0.34	FPR	0.30	0.39	0.32	0.34
FNR	0.09	0.06	0.22	0.08	FNR	0.09	0.06	0.22	0.08
Precision	0.85	0.82	0.82	0.83	Precision	0.85	0.82	0.82	0.83
Recall	0.91	0.94	0.78	0.92	Recall	0.91	0.94	0.78	0.92
F1	0.88	0.87	0.80	0.87	F1	0.88	0.87	0.80	0.87

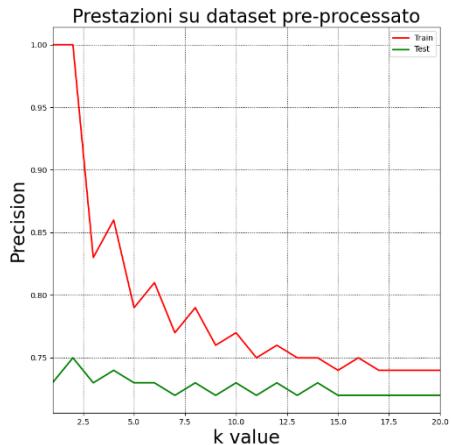
Confronto aggregazione di features

Nell'aggregazione delle feature si è scelto come numero di attributi finale tre, nonostante con la funzione implementata nel codice si possa scegliere il valore di attributi che si desidera. I risultati ottenuti mostreranno che questa scelta non conviene, poiché i valori di tutte le metriche diminuiscono notevolmente.

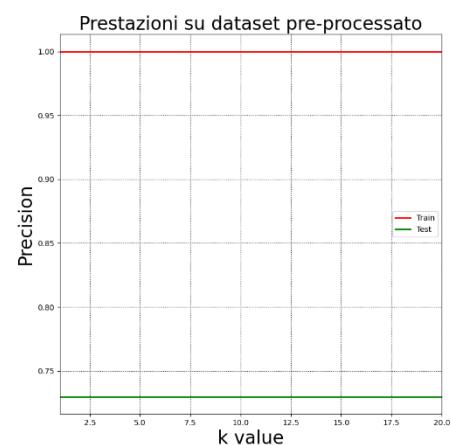
KNN

L'aggregazione di feature, svolta tramite PCA, non ha fornito dei nuovi attributi validi per la classificazione del modello k-NN. Infatti i valori delle metriche, con uno sguardo particolare nei confronti della precision, sono molto più bassi di quelle ottenute usando tutti gli attributi. Come è accaduto in molti altri casi usare le distanze pesate non ha premiato, ma al contrario porta all'overfitting del modello. Tra la distanza euclidea e quella di Manhattan, come in molti altri casi, non c'è notevole differenza. A quanto si può osservare con la PCA si perdono delle caratteristiche importanti per gli attributi, se si usa un classificatore k-NN con questo dataset.

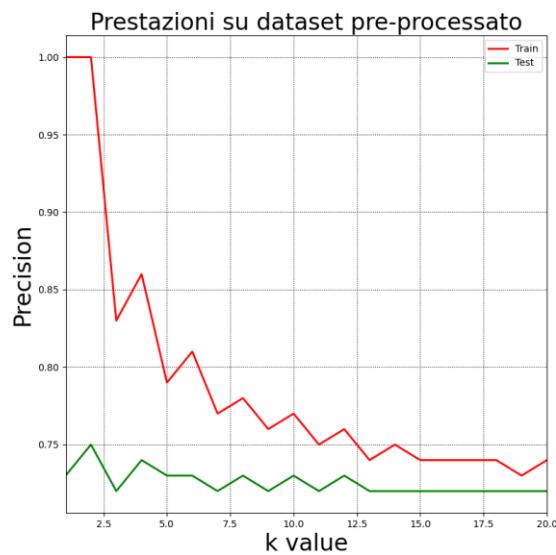
Prestazioni kNN con diversi k value



Prestazioni kNN con diversi k value con distanze pesate

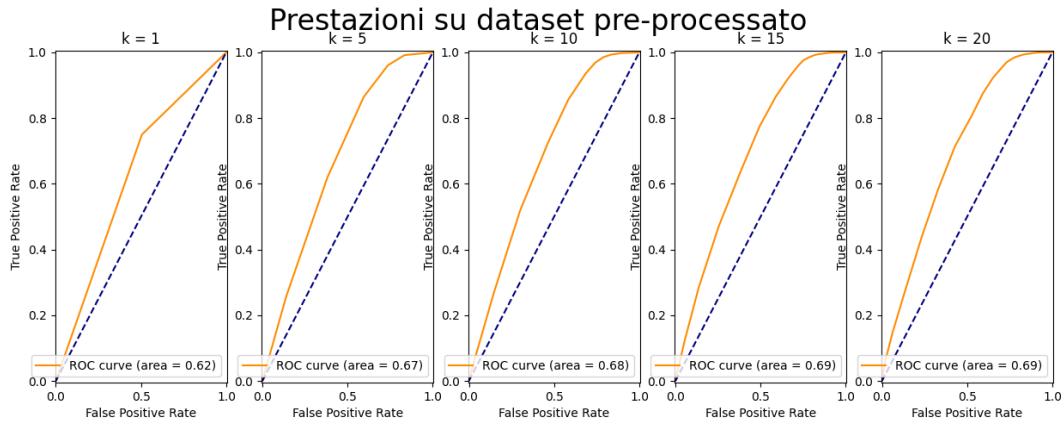


Prestazioni kNN con diversi k value con distanza di manhattan

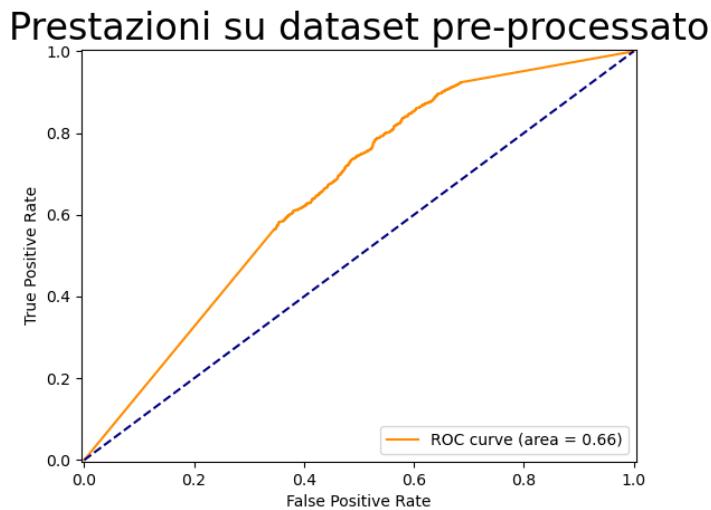


Nelle curve ROC le prestazioni sono molto vicine alla bisettrice del grafico, creando un modello molto inefficiente. Questo è correlato al fatto che anche la precision sia bassa, dimostrando che i dati aggregati sono poco efficaci.

Prestazioni kNN distanza euclidea, distanza Manhattan e distanze pesate



Prestazioni kNN tuned



Di seguito sono rappresentate tutte le metriche che sono state calcolate, e si è trovata la combinazione migliore con $k=2$, peso uniforme per tutte le distanze e utilizzo della distanza Manhattan (ultima foto).

Metriche di valutazione kNN

Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.66	0.60	0.69	0.66	0.70	0.68	0.70	0.69	0.71	0.70	0.71	0.70	0.72	0.71	0.72	0.72	0.72	0.72	0.72	0.72	
Error rate	0.34	0.40	0.31	0.34	0.30	0.32	0.30	0.31	0.29	0.30	0.29	0.30	0.28	0.29	0.28	0.29	0.28	0.28	0.28	0.28	
TP	2308.00	1733.00	2525.00	2224.00	2662.00	2428.00	2714.00	2544.00	2766.00	2637.00	2798.00	2692.00	2818.00	2749.00	2833.00	2779.00	2875.00	2825.00	2891.00	2840.00	
TPR	0.75	0.56	0.82	0.72	0.86	0.79	0.88	0.83	0.90	0.86	0.91	0.87	0.92	0.89	0.92	0.90	0.93	0.92	0.94	0.92	
TNR	0.50	0.66	0.44	0.53	0.41	0.47	0.37	0.43	0.37	0.42	0.35	0.39	0.35	0.38	0.34	0.36	0.33	0.36	0.32	0.35	
FPR	0.50	0.34	0.56	0.47	0.59	0.53	0.63	0.57	0.63	0.58	0.65	0.61	0.65	0.62	0.66	0.64	0.67	0.64	0.68	0.65	
FNR	0.25	0.44	0.18	0.28	0.14	0.21	0.12	0.17	0.10	0.14	0.09	0.08	0.11	0.08	0.10	0.07	0.08	0.06	0.08	0.08	
Precision	0.73	0.75	0.73	0.74	0.73	0.73	0.72	0.73	0.72	0.73	0.72	0.73	0.72	0.73	0.72	0.72	0.72	0.72	0.72	0.72	
Recall	0.75	0.56	0.82	0.72	0.86	0.79	0.88	0.83	0.90	0.86	0.91	0.87	0.92	0.89	0.92	0.90	0.93	0.92	0.94	0.92	
F1	0.74	0.64	0.77	0.73	0.79	0.76	0.79	0.77	0.80	0.79	0.80	0.79	0.81	0.80	0.81	0.81	0.81	0.81	0.81	0.81	
Il valore di k che porta precisione migliore è 2 che classifica 1733.0 TP con precisione 0.75 senza cambiare alcun iperparametro eccetto k																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.66	0.66	0.69	0.69	0.70	0.70	0.71	0.71	0.72	0.72	0.72	0.72	0.73	0.73	0.72	0.73	0.73	0.73	0.73	0.73	
Error rate	0.34	0.34	0.31	0.31	0.30	0.30	0.29	0.29	0.28	0.28	0.28	0.28	0.27	0.27	0.28	0.27	0.27	0.27	0.27	0.27	
TP	2308.00	2308.00	2509.00	2517.00	2613.00	2633.00	2680.00	2695.00	2738.00	2748.00	2774.00	2781.00	2797.00	2806.00	2827.00	2834.00	2838.00	2851.00	2859.00	2866.00	
TPR	0.75	0.75	0.81	0.82	0.85	0.86	0.87	0.88	0.89	0.89	0.90	0.91	0.91	0.92	0.92	0.92	0.93	0.93	0.93	0.93	
TNR	0.50	0.50	0.46	0.45	0.43	0.42	0.40	0.40	0.39	0.38	0.39	0.37	0.37	0.37	0.36	0.37	0.36	0.37	0.36	0.35	
FPR	0.50	0.50	0.54	0.55	0.57	0.58	0.60	0.60	0.60	0.61	0.62	0.61	0.62	0.63	0.63	0.64	0.63	0.64	0.65	0.65	
FNR	0.25	0.25	0.19	0.18	0.15	0.14	0.13	0.12	0.11	0.11	0.10	0.10	0.09	0.09	0.08	0.08	0.08	0.07	0.07	0.07	
Precision	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	
Recall	0.75	0.75	0.81	0.82	0.85	0.86	0.87	0.88	0.89	0.89	0.90	0.91	0.92	0.92	0.92	0.92	0.93	0.93	0.93	0.93	
F1	0.74	0.74	0.77	0.77	0.79	0.79	0.79	0.80	0.80	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.82	0.81	0.81	
Il valore di k che porta precisione migliore è 1 che classifica 2308.0 TP con precisione 0.73 pesando le distanze																					

Metriche di valutazione, test set:																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.66	0.66	0.69	0.69	0.70	0.70	0.71	0.71	0.72	0.72	0.72	0.72	0.72	0.73	0.73	0.73	0.72	0.73	0.73	0.73
Error rate	0.34	0.34	0.31	0.31	0.30	0.30	0.29	0.29	0.28	0.28	0.28	0.28	0.28	0.27	0.27	0.28	0.27	0.27	0.27	0.27
TP	2308.00	2308.00	2509.00	2517.00	2613.00	2633.00	2680.00	2695.00	2738.00	2748.00	2774.00	2781.00	2797.00	2806.00	2827.00	2834.00	2838.00	2851.00	2859.00	2866.00
TPR	0.75	0.75	0.81	0.82	0.85	0.86	0.87	0.88	0.89	0.89	0.90	0.90	0.91	0.91	0.92	0.92	0.93	0.93	0.93	0.93
TNR	0.50	0.50	0.46	0.45	0.43	0.42	0.40	0.40	0.40	0.39	0.38	0.39	0.38	0.37	0.37	0.36	0.37	0.36	0.36	0.35
FPR	0.50	0.50	0.54	0.55	0.57	0.58	0.60	0.60	0.60	0.61	0.62	0.61	0.62	0.63	0.63	0.63	0.64	0.63	0.64	0.65
FNR	0.25	0.25	0.19	0.18	0.15	0.14	0.13	0.12	0.11	0.11	0.10	0.10	0.09	0.09	0.08	0.08	0.08	0.07	0.07	0.07
Precision	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
Recall	0.75	0.75	0.81	0.82	0.85	0.86	0.87	0.88	0.89	0.89	0.90	0.90	0.91	0.91	0.92	0.92	0.93	0.93	0.93	0.93
F1	0.74	0.74	0.77	0.77	0.79	0.79	0.79	0.80	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.82	0.81	0.81	0.82

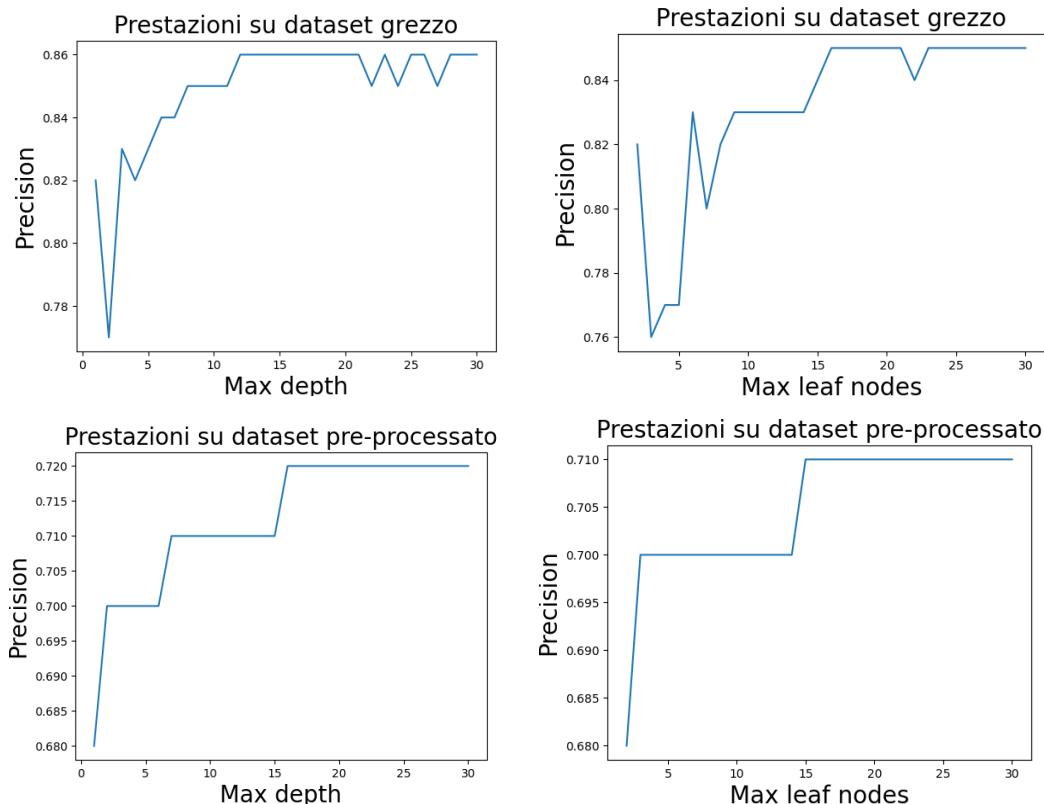
Metriche di valutazione, test set:
il valore di k che porta precisione migliore è 1 che classifica 2308.0 TP con precisione 0.73 pesando le distanze

Metriche di valutazione, test set:																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.60																			
Error rate	0.40																			
TP	1755.00																			
TPR	0.57																			
TNR	0.66																			
FPR	0.34																			
FNR	0.43																			
Precision	0.75																			
Recall	0.57																			
F1	0.65																			

Metriche di valutazione, test set:
il valore di k che porta precisione migliore è 2 che classifica 1755.0 TP con precisione 0.75 con k=2, weight=uniform e p=1

Dtree

Negli alberi decisionali, contrariamente al k-NN, le prestazioni con il dataset aggregato tramite PCA migliorano nei primi valori, ma alla lunga l'andamento risulta simile a quello sui dati non processati, con l'unica differenza che i dati processati hanno un andamento meno spigoloso.



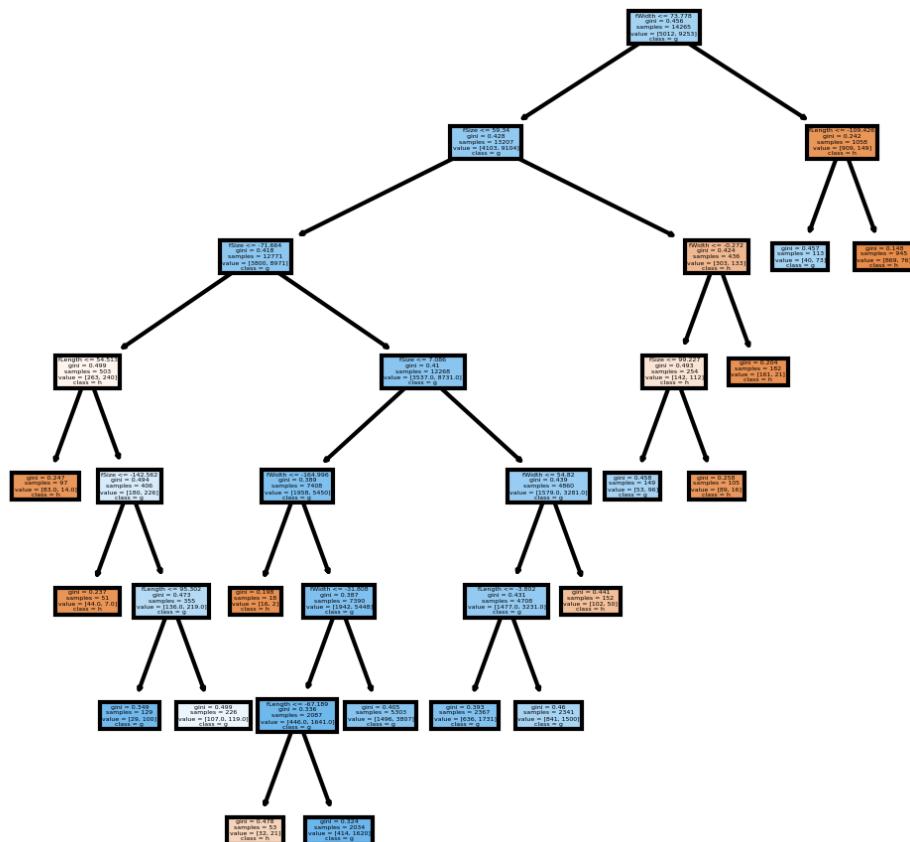
Nell'immagine seguente sono mostrate prima le metriche dell'albero tuned e poi quelle degli alberi generati facendo variare l'iperparametro del min_gain. Si può notare come, anche in questo caso, esse siano inferiori rispetto alle metriche ottenute sui dati grezzi.

Metriche di valutazione dTree

Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:						
	Tuned					
Accuracy	0.72					
Error rate	0.28					
TP	3005.00					
TPR	0.98					
TNR	0.25					
FPR	0.75					
FNR	0.02					
Precision	0.71					
Recall	0.98					
F1	0.82					
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:						
	0.1000	0.0500	0.0300	0.0080	0.0075	0.0050
Accuracy	0.65	0.65	0.69	0.71	0.71	0.71
Error rate	0.35	0.35	0.31	0.29	0.29	0.29
TP	3079.00	3079.00	3029.00	2985.00	2985.00	2985.00
TPR	1.00	1.00	0.98	0.97	0.97	0.97
TNR	0.00	0.00	0.16	0.23	0.23	0.23
FPR	1.00	1.00	0.84	0.77	0.77	0.77
FNR	0.00	0.00	0.02	0.03	0.03	0.03
Precision	0.65	0.65	0.68	0.70	0.70	0.70
Recall	1.00	1.00	0.98	0.97	0.97	0.97
F1	0.79	0.79	0.81	0.81	0.81	0.81

Le immagini seguenti rappresentano gli alberi decisionali ottenuti dai criteri usati in precedenza.

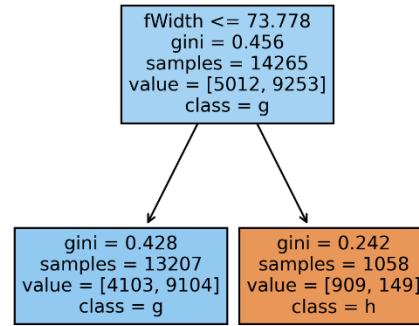
Albero tuned dataset pre-processato



Tree pre-processato con mingain 0.1

gini = 0.456
samples = 14265
value = [5012, 9253]
class = g

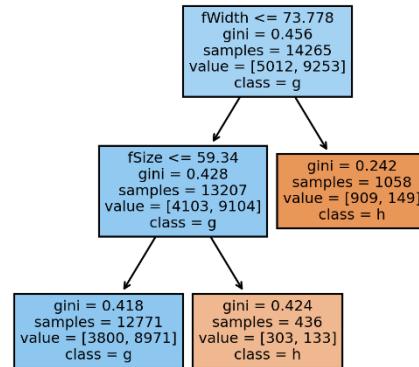
Tree pre-processato con mingain 0.03



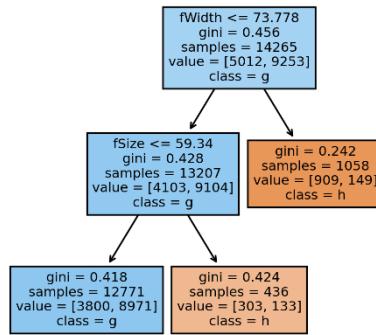
Tree pre-processato con mingain 0.05

gini = 0.456
samples = 14265
value = [5012, 9253]
class = g

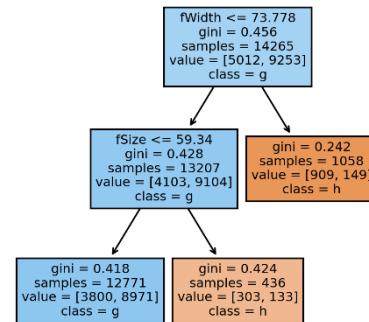
Tree pre-processato con mingain 0.005



Tree pre-processato con mingain 0.008



Tree pre-processato con mingain 0.0075



Classificatore multiplo

Anche usando il classificatore multiplo le prestazioni peggiorano secondo tutte le metriche, sempre a causa dei modelli ad alberi decisionali che sono utilizzati come classificatori.

Prestazioni sul dataset processato					Prestazioni sul dataset grezzo				
	Clf_1	Clf_2	Clf_3	Clf_final		Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.72	0.71	0.65	0.71	Eseguo classificatore multiplo custom su dataset grezzo...				
Error rate	0.28	0.29	0.35	0.29	Metriche di valutazione, test set, su dataset grezzo:				
TP	3005.00	2985.00	3079.00	3032.00	Clf_1	Clf_2	Clf_3	Clf_final	
TPR	0.98	0.97	1.00	0.98	Accuracy	0.83	0.82	0.74	0.83
TNR	0.25	0.23	0.00	0.20	Error rate	0.17	0.18	0.26	0.17
FPR	0.75	0.77	1.00	0.80	TP	2796.00	2885.00	2392.00	2831.00
FNR	0.02	0.03	0.00	0.02	TPR	0.91	0.94	0.78	0.92
Precision	0.71	0.70	0.65	0.69	TNR	0.70	0.61	0.68	0.66
Recall	0.98	0.97	1.00	0.98	FPR	0.30	0.39	0.32	0.34
F1	0.82	0.81	0.79	0.81	FNR	0.09	0.06	0.22	0.08

Confronto selezione di features

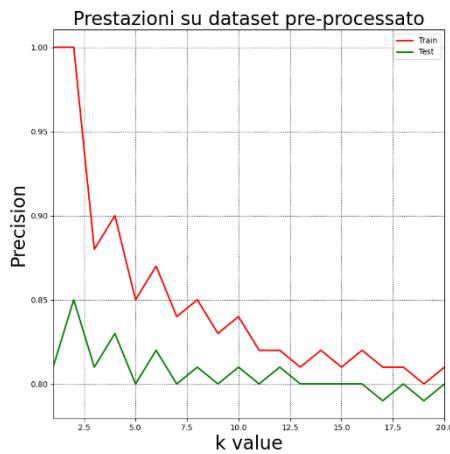
Nella selezione di feature sono stati menzionati in precedenza i criteri sulla scelta degli attributi da escludere, ovvero i tre attributi dalla varianza più bassa e di conseguenza con meno valori unici.

KNN

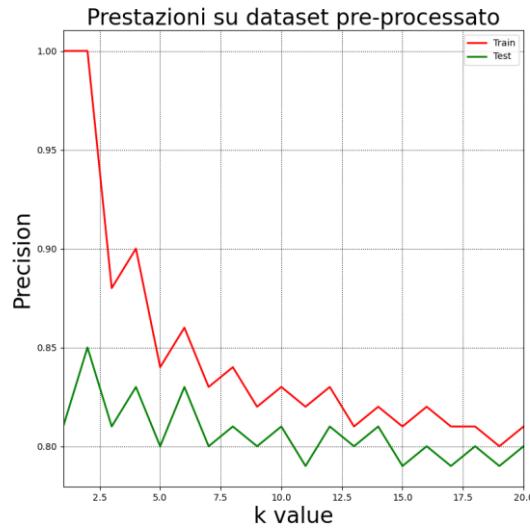
Nel classificatore k-NN le prestazioni sono notevolmente migliori rispetto a quelle ottenute con PCA, probabilmente perché mantengono delle caratteristiche più importanti. Tra la distanza euclidea e la distanza Manhattan non vi è molta differenza, come in molti casi precedenti, mentre pesando le distanze si va incontro al solito problema di overfitting che diminuisce la precision. In generale le prestazioni eliminando tre attributi non variano notevolmente, e proprio per questo converrebbe usare questo dataset ridotto rispetto a quello originale.

Prestazioni kNN con diversi k value

Prestazioni kNN con diversi k value con distanze pesate

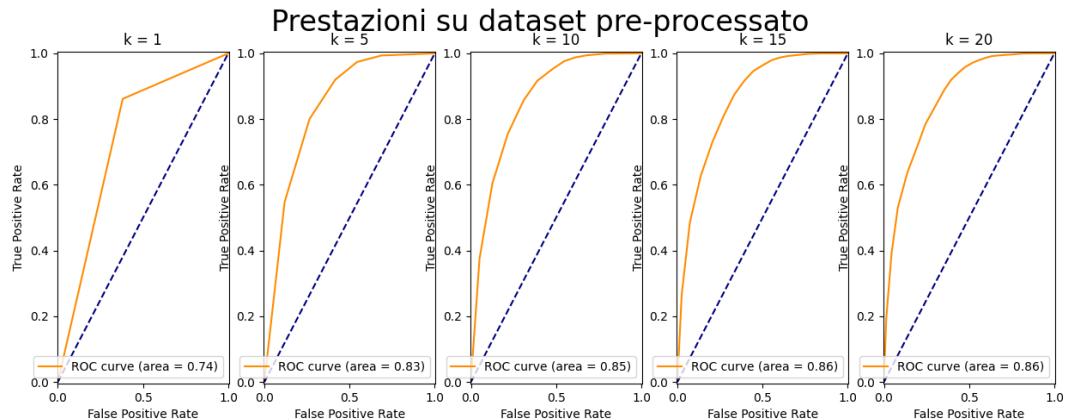


Prestazioni kNN con diversi k value con distanza di manhattan

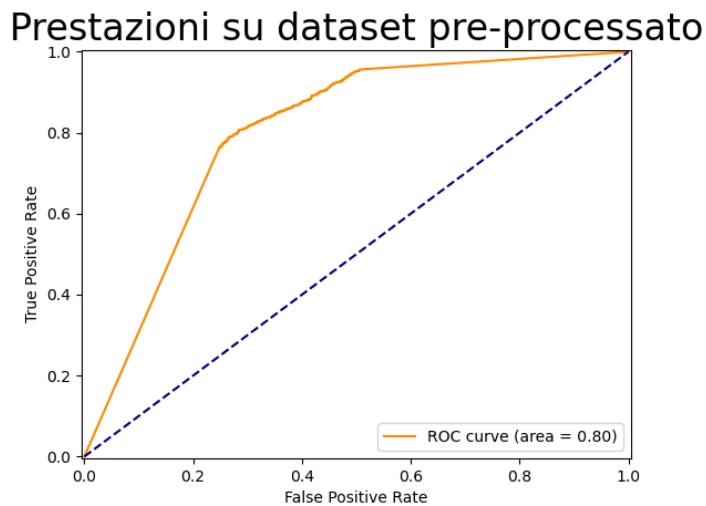


Per quanto riguarda le curve ROC le AUC, anche queste sono paragonabili a quelle ottenute prima della selezione degli attributi.

Prestazioni kNN distanza euclidea, distanza Manhattan e distanze pesate



Prestazioni kNN tuned



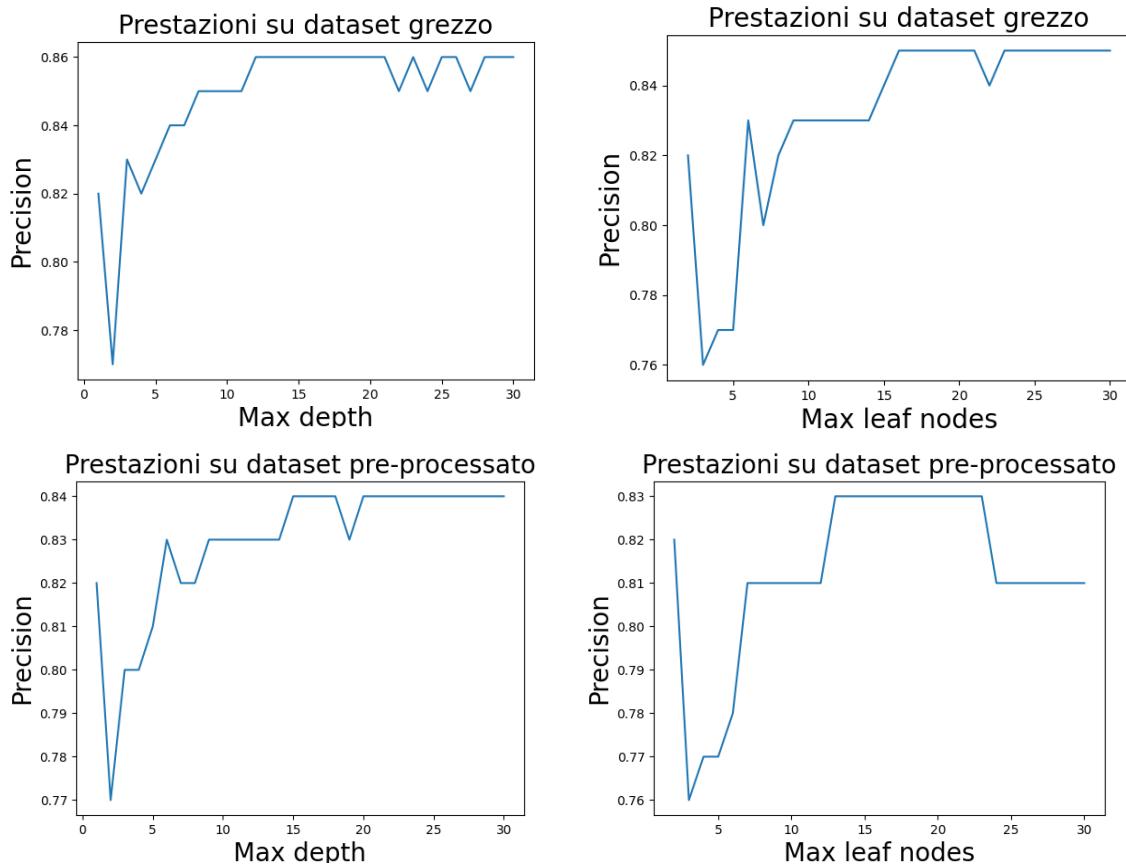
Di seguito sono mostrati i valori delle metriche ottenute usando distanza euclidea, pesando le distanze e usando la distanza Manhattan. Il modello ideale dopo il tuning degli iperparametri è quello che utilizza k=2, senza pesare le distanze (per non incorrere nell'overfitting) e usando la distanza Manhattan.

Metriche di valutazione kNN

Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.78	0.76	0.80	0.79	0.80	0.80	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	
Error rate	0.22	0.24	0.20	0.21	0.20	0.20	0.20	0.20	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	
TP	2653.00	2346.00	2771.00	2609.00	2831.00	2734.00	2867.00	2789.00	2880.00	2822.00	2890.00	2849.00	2903.00	2871.00	2914.00	2875.00	2918.00	2886.00	2918.00	2894.00	
TPR	0.86	0.76	0.90	0.85	0.92	0.89	0.93	0.91	0.94	0.92	0.94	0.93	0.94	0.93	0.95	0.93	0.95	0.94	0.95	0.94	
TNR	0.62	0.75	0.60	0.69	0.58	0.65	0.57	0.62	0.57	0.61	0.56	0.60	0.56	0.58	0.55	0.58	0.54	0.57	0.54	0.57	
FPR	0.38	0.25	0.40	0.31	0.42	0.35	0.43	0.38	0.43	0.39	0.44	0.40	0.44	0.42	0.45	0.42	0.46	0.43	0.46	0.43	
FNR	0.14	0.24	0.10	0.15	0.08	0.11	0.07	0.09	0.06	0.08	0.06	0.07	0.06	0.07	0.05	0.07	0.05	0.06	0.05	0.06	
Precision	0.81	0.85	0.81	0.83	0.80	0.82	0.80	0.81	0.80	0.81	0.80	0.81	0.80	0.80	0.80	0.80	0.79	0.80	0.79	0.80	
Recall	0.86	0.76	0.90	0.85	0.92	0.89	0.93	0.91	0.94	0.92	0.94	0.93	0.94	0.95	0.95	0.95	0.94	0.95	0.95	0.94	
F1	0.83	0.80	0.85	0.84	0.86	0.85	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	
il valore di k che porta precisione migliore è 2 che classifica 2346.0 TP con precisione 0.85 senza cambiare alcun iperparametro eccetto k																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.78	0.78	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	
Error rate	0.22	0.22	0.20	0.20	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	
TP	2653.00	2653.00	2767.00	2785.00	2829.00	2842.00	2864.00	2871.00	2884.00	2881.00	2884.00	2895.00	2898.00	2908.00	2914.00	2910.00	2917.00	2915.00	2917.00	2918.00	
TPR	0.86	0.86	0.90	0.90	0.92	0.92	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95	
TNR	0.62	0.62	0.62	0.62	0.60	0.60	0.59	0.59	0.59	0.59	0.58	0.58	0.57	0.57	0.57	0.57	0.57	0.56	0.56	0.56	
FPR	0.38	0.38	0.38	0.38	0.40	0.40	0.41	0.41	0.41	0.41	0.42	0.42	0.43	0.43	0.43	0.43	0.43	0.44	0.44	0.44	
FNR	0.14	0.14	0.10	0.10	0.08	0.08	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.05	
Precision	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.80	0.81	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	
Recall	0.86	0.86	0.90	0.90	0.92	0.92	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95	0.95	
F1	0.83	0.83	0.85	0.86	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	
il valore di k che porta precisione migliore è 1 che classifica 2653.0 TP con precisione 0.81 pesando le distanze																					
Metriche di valutazione, test set:		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Accuracy	0.78	0.76	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	
Error rate	0.22	0.24	0.20	0.20	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	
TP	2678.00	2344.00	2807.00	2672.00	2870.00	2770.00	2894.00	2821.00	2910.00	2862.00	2916.00	2876.00	2928.00	2893.00	2938.00	2912.00	2948.00	2924.00	2949.00	2933.00	
TPR	0.87	0.76	0.91	0.87	0.93	0.90	0.94	0.92	0.95	0.93	0.95	0.93	0.95	0.94	0.95	0.95	0.96	0.95	0.96	0.95	
TNR	0.62	0.75	0.60	0.67	0.58	0.65	0.58	0.62	0.57	0.60	0.55	0.59	0.55	0.58	0.54	0.57	0.57	0.56	0.54	0.56	
FPR	0.38	0.25	0.40	0.33	0.42	0.35	0.42	0.38	0.43	0.40	0.45	0.41	0.45	0.42	0.46	0.43	0.46	0.44	0.46	0.44	
FNR	0.13	0.24	0.09	0.13	0.07	0.10	0.06	0.08	0.05	0.07	0.05	0.07	0.05	0.06	0.05	0.05	0.04	0.05	0.04	0.05	
Precision	0.81	0.85	0.81	0.83	0.80	0.83	0.80	0.81	0.80	0.81	0.79	0.81	0.80	0.81	0.79	0.80	0.79	0.80	0.79	0.80	
Recall	0.87	0.76	0.91	0.87	0.93	0.90	0.94	0.92	0.95	0.93	0.95	0.93	0.95	0.94	0.95	0.95	0.96	0.95	0.96	0.95	
F1	0.84	0.80	0.86	0.85	0.86	0.86	0.87	0.86	0.87	0.86	0.87	0.87	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	
il valore di k che porta precisione migliore è 2 che classifica 2344.0 TP con precisione 0.85 utilizzando la distanza Manhattan																					
Metriche di valutazione, test set:		2																			
Accuracy		0.76																			
Error rate		0.24																			
TP		2344.00																			
TPR		0.76																			
TNR		0.75																			
FPR		0.25																			
FNR		0.24																			
Precision		0.85																			
Recall		0.76																			
F1		0.80																			
il valore di k che porta precisione migliore è 2 che classifica 2344.0 TP con precisione 0.85 con k=2, weight=uniform e p=1																					

Dtree

Anche per l'albero decisionale non si nota una differenza eccessiva tra i dati testati a partire dal dataset grezzo o dal dataset processato, perciò conviene utilizzare il secondo che comporta un costo computazionale inferiore.

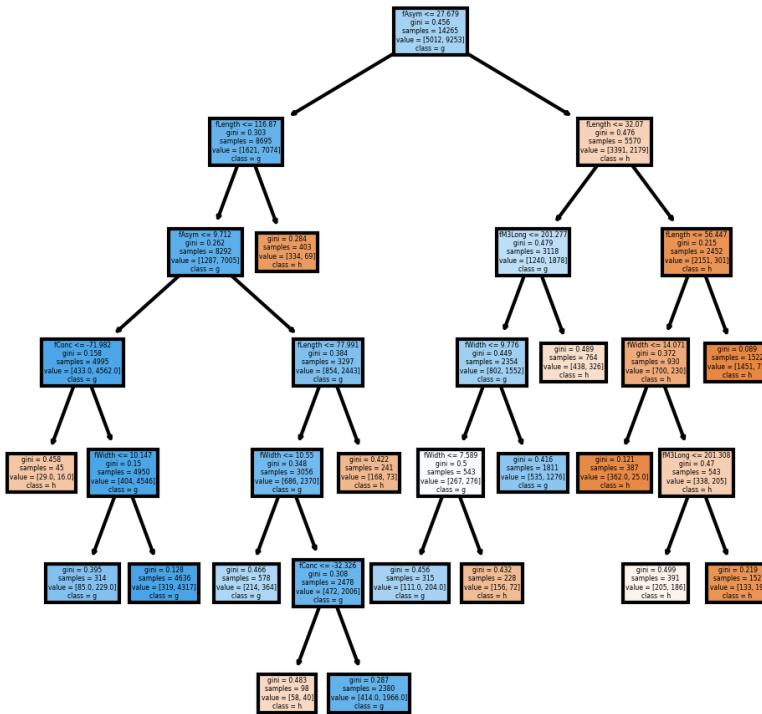


Nell'immagine seguente sono mostrate prima le metriche dell'albero tuned e poi quelle degli alberi generati facendo variare l'iperparametro min_gain.

Metriche di valutazione dTree						
Metriche di valutazione per il test set, con albero tuned, su dataset pre-processato:						
	Tuned					
Accuracy	0.81					
Error rate	0.19					
TP	2760.00					
TPR	0.90					
TNR	0.66					
FPR	0.34					
FNR	0.10					
Precision	0.83					
Recall	0.90					
F1	0.86					
Metriche di valutazione per il test set relativo al minimo guadagno, su dataset pre-processato:						
	0.1000	0.0500	0.0300	0.0080	0.0075	0.0050
Accuracy	0.65	0.74	0.78	0.79	0.79	0.80
Error rate	0.35	0.26	0.22	0.21	0.21	0.20
TP	3079.00	2392.00	2984.00	2963.00	2963.00	2918.00
TPR	1.00	0.78	0.97	0.96	0.96	0.95
TNR	0.00	0.68	0.43	0.48	0.48	0.52
FPR	1.00	0.32	0.57	0.52	0.52	0.48
FNR	0.00	0.22	0.03	0.04	0.04	0.05
Precision	0.65	0.82	0.76	0.77	0.77	0.78
Recall	1.00	0.78	0.97	0.96	0.96	0.95
F1	0.79	0.80	0.85	0.86	0.86	0.86

Le immagini seguenti rappresentano gli alberi decisionali ottenuti dai criteri usati in precedenza.

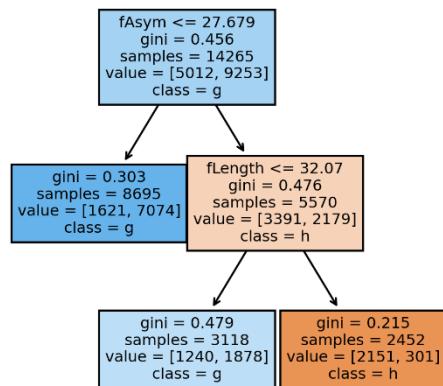
Albero tuned dataset pre-processato



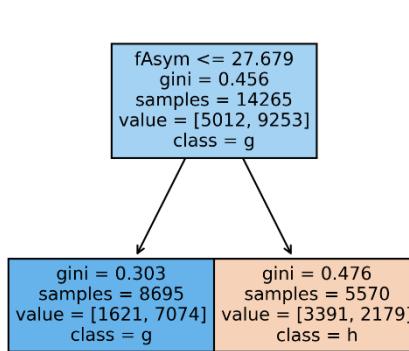
Tree pre-processato con mingain 0.1

Tree pre-processato con mingain 0.03

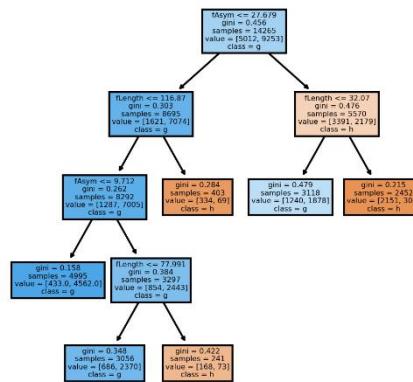
**gini = 0.456
samples = 14265
value = [5012, 9253]
class = g**



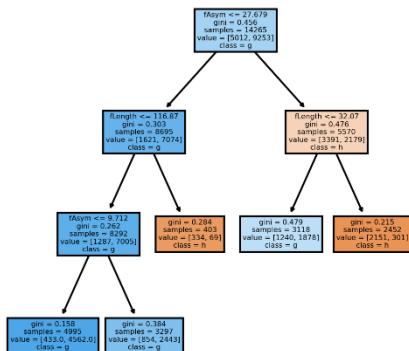
Tree pre-processato con mingain 0.05



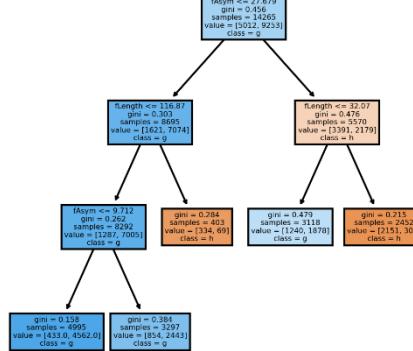
Tree pre-processato con mingain 0.005



Tree pre-processato con mingain 0.008



Tree pre-processato con mingain 0.0075



Classificatore multiplo

Nel classificatore multiplo le prestazioni riguardo la metrica precision sono lievemente inferiori rispetto al dataset non processato, a causa del numero inferiore di feature che sono state utilizzate, guadagnando in compenso un modello più semplice.

Prestazioni sul dataset processato

	Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.81	0.80	0.74	0.81
Error rate	0.19	0.20	0.26	0.19
TP	2760.00	2918.00	2392.00	2785.00
TPR	0.90	0.95	0.78	0.90
TNR	0.66	0.52	0.68	0.64
FPR	0.34	0.48	0.32	0.36
FNR	0.10	0.05	0.22	0.10
Precision	0.83	0.78	0.82	0.82
Recall	0.90	0.95	0.78	0.90
F1	0.86	0.86	0.80	0.86

Prestazioni sul dataset grezzo

	Clf_1	Clf_2	Clf_3	Clf_final
Accuracy	0.83	0.82	0.74	0.83
Error rate	0.17	0.18	0.26	0.17
TP	2796.00	2885.00	2392.00	2831.00
TPR	0.91	0.94	0.78	0.92
TNR	0.70	0.61	0.68	0.66
FPR	0.30	0.39	0.32	0.34
FNR	0.09	0.06	0.22	0.08
Precision	0.85	0.82	0.82	0.83
Recall	0.91	0.94	0.78	0.92
F1	0.88	0.87	0.80	0.87

5. Conclusione e scelta del classificatore migliore

Come conclusione verrà presentata la combinazione di classificatore, tuning degli iperparametri riferiti al classificatore e tecnica di pre processing che si sono rivelati migliori per la task preposta. Si ricorda che l'obiettivo era identificare la classe g , considerata positiva, tra un background di classe h , considerata negativa. Si è deciso di avere un'elevato livello di sicurezza che ogni record classificato come g lo sia effettivamente, questo perché si è immaginata che degli studi condotti sui raggi gamma abbiano necessità di basarsi su dati certi. Questo anche a scapito dell'accuratezza generale del classificatore. Si tratta quindi di indurre un classificatore molto specifico, basato sul massimizzare la misura Precision per la classificazione dei TP.

Al contrario di quello che è stato visto con i dati grezzi, con i dati pre processati il classificatore basato su istanze kNN offre prestazioni migliori rispetto al classificatore logico albero decisionale. Per questo, anche rispetto al classificatore multiplo, che ha comunque una base composta da una terna di alberi decisionali, è stato scelto come migliore un classificatore kNN.

Il kNN in questione è quello definito tuned: l'analisi delle prestazioni sui tre iperparametri k , weights e p , (ovvero il numero di vicini da utilizzare per la classificazione, l'eventuale peso da dare sulla base della distanza, e il tipo di distanza utilizzata) ha portato alla scelta di $k=2$, di non pesare le distanze e di utilizzare la distanza di tipo Manhattan. Questo significa che la complessità computazionale, che generalmente con un classificatore basato su istanze è particolarmente alta, è invece estremamente ridotta: si utilizzano soltanto due vicini per la classificazione, non vengono utilizzati dei pesi e la distanza calcolata è la più semplice possibile: non è necessario calcolare valori al quadrato e radici quadrate.

Il kNN tuned raggiunge prestazioni degne di nota non con i dati grezzi, ma con un particolare dataset pre processato, basato sull'oversampling.

Ci si aspettava che i modelli funzionassero meglio con tecniche di oversampling rispetto a tecniche di undersampling, questo perché la classe g , maggioritaria nei dati grezzi, è il dato più prezioso ed è stato ritenuto preferibile aumentare artificialmente gli esempi di classe h , piuttosto che ridurre gli esempi di classe g . Questa credenza è stata in parte smentita da prestazioni di modelli come gli alberi decisionali, ma viene rispettata per quanto riguarda il classificatore migliore.

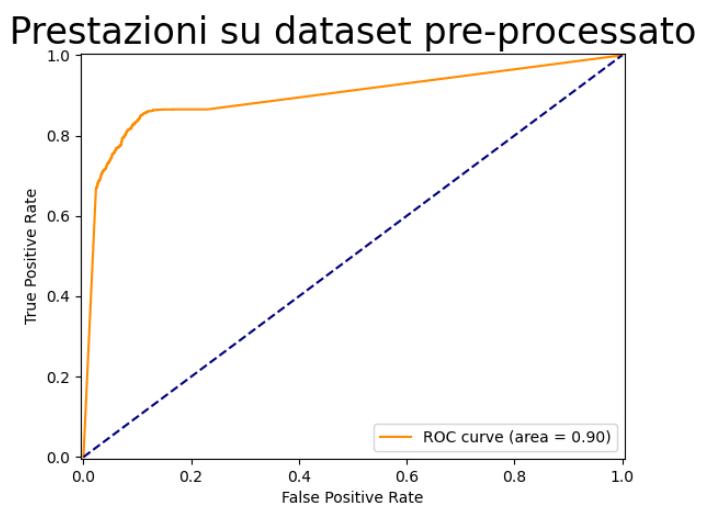
Esso viene indotto su un dataset al quale è stato applicato l'algoritmo di oversampling ADASYN, che si basa sulla generazione di record artificiali nelle vicinanze dei campioni originali classificati erroneamente con un classificatore kNN. Questo processo permette di migliorare notevolmente la decision boundary, portando ai risultati migliori ottenuti tra tutte le combinazioni.

Qui di seguito vengono mostrate le metriche del classificatore: si ottiene una precisione di ben **0.97** classificando **1596 TP**.

Nonostante il numero dei TP sia minore rispetto ad altri classificatori, nessuno regge il confronto per quanto riguarda la precisione.

E' da notare che, calcolando la curva ROC del classificatore, essa ha un'AUC di ben 0.90, rendendolo un classificatore molto buono anche a livello generale.

Qua di seguito vengono visualizzate la curva ROC e le corrispondenti metriche del classificatore.



```
Metriche di valutazione, test set:
    2
Accuracy      0.82
Error rate    0.18
TP            1569.00
TPR           0.67
TNR           0.98
FPR           0.02
FNR           0.33
Precision     0.97
Recall        0.67
F1            0.79
il valore di k che porta precisione migliore è 2 che classifica 1569.0 TP con precisione 0.97 con k=2, weight=uniform e p=1
```