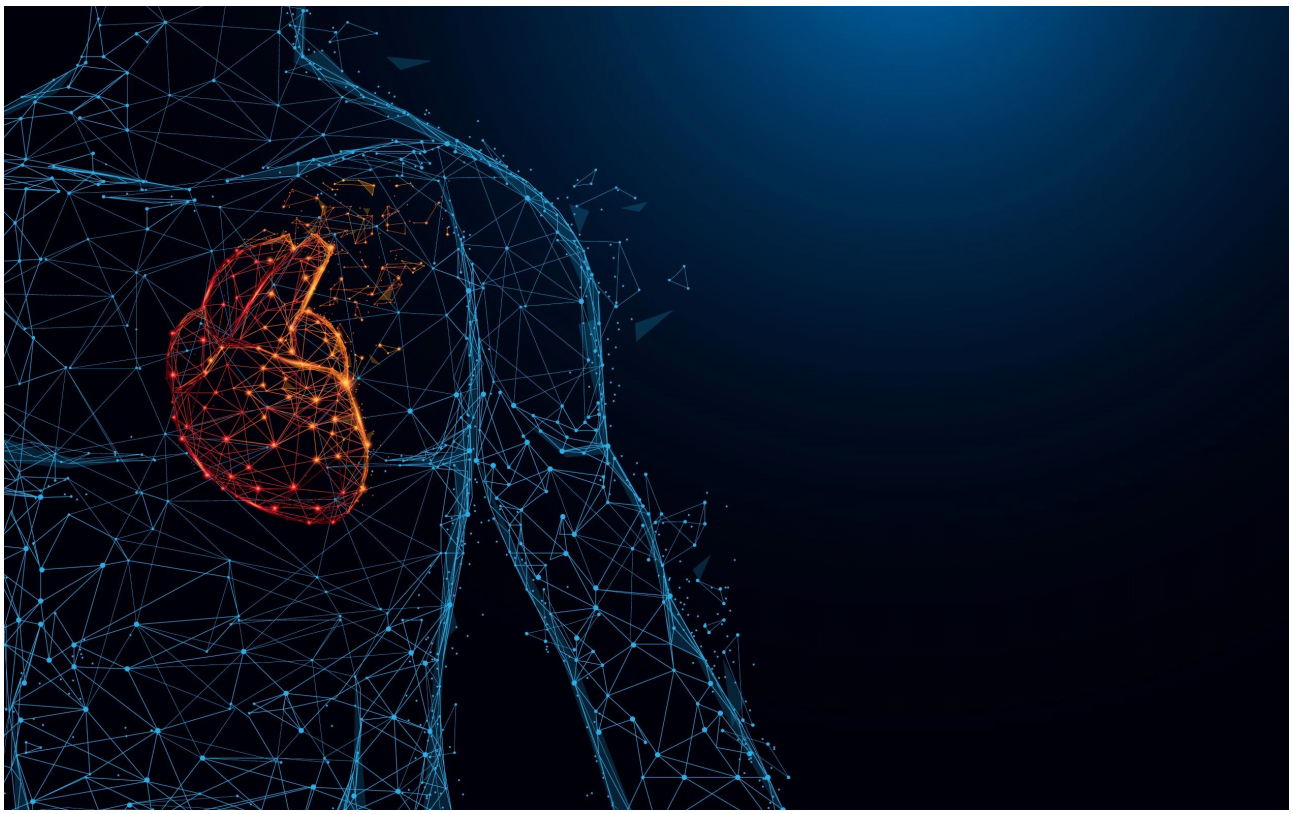
PROGETTO INGEGNERIA DELLA CONOSCENZA

Autore: Federico Chiaradia

Email: f.chiaradia5@studenti.uniba.it

Matricola: 697798

Repository: https://github.com/Fredbcx/Progetto-Icon21



Indice

Introduzione

Il progetto prevede l'utilizzo di tecniche di apprendimento automatico con lo scopo di prevedere la sopravvivenza di pazienti aventi insufficienza cardiaca, per mezzo di classificatori.

Il Dataset utilizzato contiene 13 caratteristiche cliniche di 299 pazienti con insufficienza cardiaca, raccolte durante il periodo di follow-up (monitoraggio a seguito di un'azione o intervento).
Fonte: https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records

Features:

Età: età del paziente (anni)

Anemia: diminuzione dei globuli rossi o dell'emoglobina (booleano)

Pressione alta: se il paziente ha ipertensione (booleano)

Creatina Fosfochinasi (CPK)[1]: livello dell'enzima CPK nel sangue (mcg/L)

Diabete: se il paziente ha il diabete (booleano)

Frazione di eiezione: percentuale di sangue che esce dal cuore ad ogni contrazione (percentuale)

Piastrine: piastrine nel sangue (kilopiastrine/mL)

Sesso: donna o uomo (binario)

Creatinina sierica[2]: livello di creatinina sierica nel sangue (mg/dL)

Sodio sierico[3]: livello di sodio sierico nel sangue (mEq/L)

Fumatore: se il paziente fuma o meno (booleano)

Tempo: periodo di follow-up (giorni)

Evento di morte [target]: se il paziente è deceduto durante il periodo di follow-up (booleano)

1: La creatinchinasi (CK) o creatina fosfochinasi (CPK) è un enzima presente soprattutto nel tessuto muscolare scheletrico e nelle fibre cardiache.
Il suo compito principale è quello di “facilitare” alcune reazioni chimiche, che avvengono fisiologicamente nel nostro organismo. Più nel dettaglio, la creatinchinasi permette la conversione della creatina in fosfocreatina, in modo tale da consumare ATP e generare energia altamente sfruttabile.2

2: La creatinina è una sostanza che deriva dalla creatina. Viene liberata nel sangue a seguito di lavoro muscolare o di un danno muscolare e il suo valore è un indicatore del funzionamento della filtrazione renale.

3: Il sodio è un importante elemento chimico extracellulare che regola la distribuzione dell´ acqua (osmolarità plasmatica) nell´ organismo umano. I livelli di sodio nel sangue sono il risultato tra la quantità ingerita con il cibo e quella eliminata attraverso i reni.

Analizzando un set di dati di 299 pazienti con insufficienza cardiaca, vengono applicati diversi classificatori di apprendimento automatico per predire la sopravvivenza dei pazienti, e altri metodi per trovare le feature più rilevanti come fattori di rischio ed essere utilizzate come strumenti per la classificazione.

NOTA: Il progetto in questione è proposto per osservare la tematica trattata da un punto di vista tecnico, inerente al corso di Ingegneria della conoscenza, pertanto non vengono effettuate considerazioni e applicazioni di nozioni biostatistiche.

Apprendimento supervisionato e non supervisionato

L'apprendimento supervisionato è una tecnica di apprendimento automatico che mira ad istruire un sistema in modo da consentirgli di elaborare automaticamente previsioni sui valori di uscita di un sistema, rispetto ad un input sulla base di una serie di esempi ideali, costituiti da coppie di input e di output, che gli vengono inizialmente forniti. Con Supervisione si intende che nell’insieme degli esempi, i segnali di output desiderati sono già noti poiché precedentemente etichettati.

L'apprendimento non supervisionato è una tecnica di apprendimento automatico che consiste nel fornire al sistema informatico una serie di input (esperienza del sistema) che egli riclassificherà ed organizzerà sulla base di caratteristiche comuni per cercare di effettuare ragionamenti e previsioni sugli input successivi. Al contrario dell'apprendimento supervisionato, durante l'apprendimento vengono forniti all'apprendista solo esempi non annotati, in quanto le classi non sono note a priori ma devono essere apprese automaticamente.

Il progetto è stato realizzato mediante il linguaggio di programmazione python, con l'ausilio della libreria Scikit-learn per gli algoritmi di apprendimento.

Di seguito vengono elencati gli algoritmi utilizzati, e successivamente riportati i risultati ottenuti da ciascuno di essi e confrontati tra loro.

- Support Vector Machine
- KNN
- Decision Tree
- Random Forest
- Multinomial Naive-Bayes
- K-Means

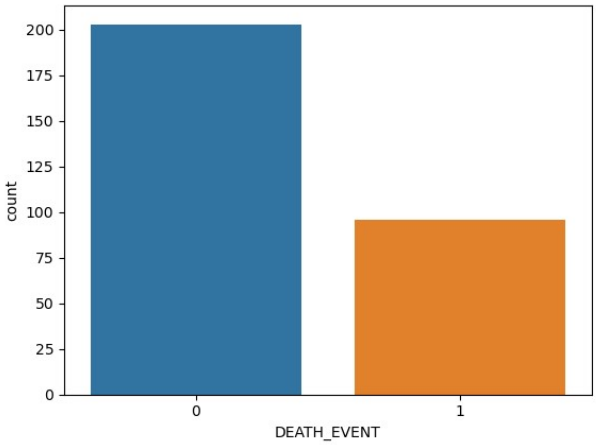
Per gli algoritmi precedentemente riportati sono stati realizzati i seguenti grafici:

- Curva ROC
  Le curve ROC riassumono il compromesso tra il tasso di veri positivi e il tasso di falsi positivi per un modello predittivo utilizzando diverse soglie di probabilità.
- Matrice di confusione
  Una matrice di confusione è una tabella in cui le previsioni sono rappresentate nelle colonne e lo stato effettivo è rappresentato dalle righe. Da questa tabella, quindi, è possibile comprendere le perf
- Curva Precisione-Recall
  Le curve Precision-Recall riassumono il compromesso tra il tasso vero positivo e il valore predittivo positivo per un modello predittivo utilizzando diverse soglie di probabilità.
- Bar Chart di varianza e deviazione standard
  La varianza rappresenta la media dei quadrati delle deviazioni, la deviazione standard è la radice quadrata del valore numerico ottenuto durante il calcolo della varianza.
- Feature importance Bar Chart

Feature Selection

La feature selection è il processo in cui automaticamente o manualmente avviene una selezione di un sottoinsieme di caratteristiche rilevanti che contribuiscono maggiormente all'output da predire.

Solitamente viene utilizzata per ridurre le dimensioni dei dati a disposizione, rimuovendo elementi irrilevanti o ridondanti, in modo semplificare i modelli e ridurre i costi. In questo progetto si utilizza la feature selction per effettuare una cernita delle caratteristiche più rilevanti rispetto alla predizione dell'evento morte, ed utilizzare il sottoinsieme ricavato per altri modelli di classificazione, che verranno poi confrontati.



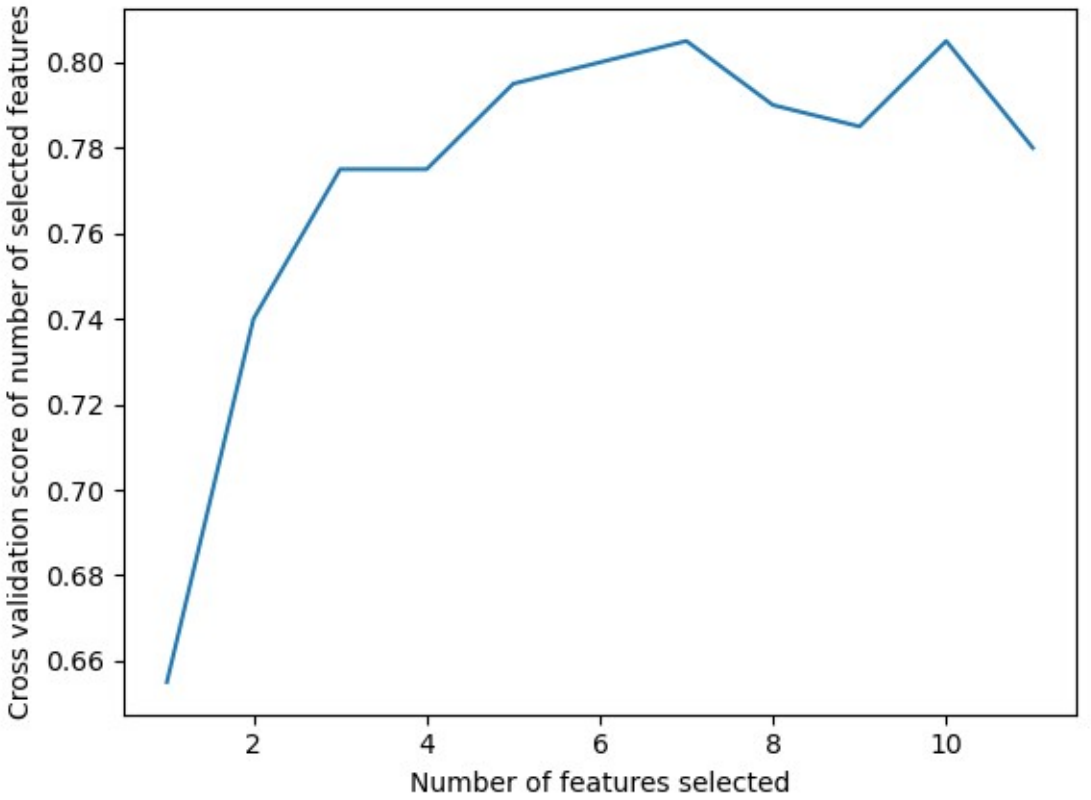
- Conteggio evento morte

Dal grafico precedente si può notare come sia sbilanciato l'evento morte.

Un problema con la classificazione sbilanciata è che ci sono troppo pochi esempi della classe di minoranza perché un modello possa apprendere efficacemente il confine decisionale.

Un modo per risolvere questo problema è "sovracampionare" gli esempi nella classe di minoranza. Ciò può essere ottenuto semplicemente duplicando esempi dalla classe di minoranza nel set di dati di addestramento prima di adattare un modello. Questo può bilanciare la distribuzione della classe ma non fornisce alcuna informazione aggiuntiva al modello.

L'approccio più diffuso per sintetizzare nuovi esempi è chiamato **SMOTE** (Synthetic Minority Over-sampling Technique), il quale funziona selezionando esempi vicini nello spazio delle feature, disegnando una linea tra gli esempi nello spazio delle caratteristiche e disegnando un nuovo campione in un punto lungo quella linea.



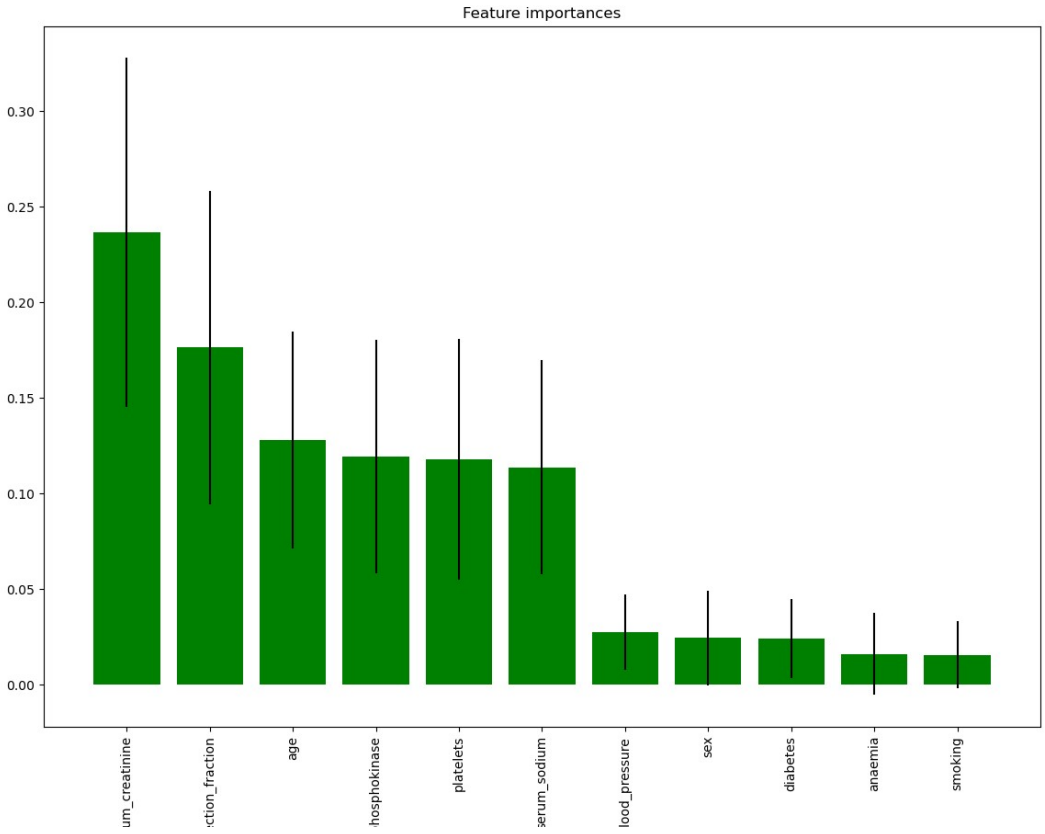
- Recursive feature elimination con cross validation e classificazione random forest

Dal grafico precedente, si può osservare come il numero di feature utilizzate per la classificazione vada ad influire sulla predizione, mediante cross validation e algoritmo random forest. Per le classificazioni successive verranno utilizzate le 2 migliori feature.

Recursive Feature Elimination (RFE)

L'obiettivo del RFE è selezionare le feature considerando ricorsivamente insiemi di feature sempre più piccoli. In primo luogo, lo stimatore viene addestrato sull'insieme iniziale di feature assegnando un'importanza a ciascuna feature. Le feature meno importanti vengono eliminate dal set di feature corrente. Tale procedura viene ripetuta in modo ricorsivo sul set potato fino a quando non viene raggiunto il numero desiderato di feature da selezionare.

Dai risultati ottenuti le feature selezionate sono Frazione di eiezione e Creatinina sierica:



Queste feature sono state successivamente utilizzate per effettuare classificazione con gli algoritmi che sono risultati più performanti e con l'algoritmo di Logistic Regression.

Utilizzando train_test_split() dalla libreria scikit-learn, è stato diviso il dataset in sottoinsiemi che riducono al minimo il potenziale di bias nel processo di valutazione e convalida, utilizzando il 30% del dataset come testset.

Questa suddivisione è stata utilizzata per ricavare i valori di punteggio medio (cross-val-score), della varianza, deviazione standard su cinque iterazioni di cross validation, per poter rilevare possibili problemi di sovra-adattamento.

Decision Tree

Gli alberi decisionali (DT) sono un metodo di apprendimento supervisionato non parametrico utilizzato per la classificazione e la regressione. L'obiettivo è creare un modello che preveda il valore di una variabile target apprendendo semplici regole decisionali dedotte dalle caratteristiche dei dati. È un modello predittivo, dove ogni nodo interno rappresenta una variabile, un arco verso un nodo figlio rappresenta un possibile valore per quella proprietà e una foglia il valore predetto per la variabile obiettivo a partire dai valori delle altre proprietà.

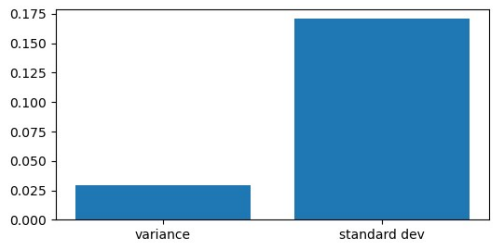
Cross Validation

Pre-SMOTE

cv_scores mean:0.6388700564971751

cv_score variance:0.029122621213572085

cv_score dev standard:0.17065351216301433

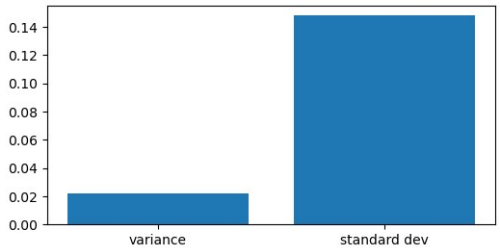


Post-SMOTE

cv_scores mean:0.7098163203854261

cv_score variance:0.021909226740566105

cv_score dev standard:0.1480176568540595



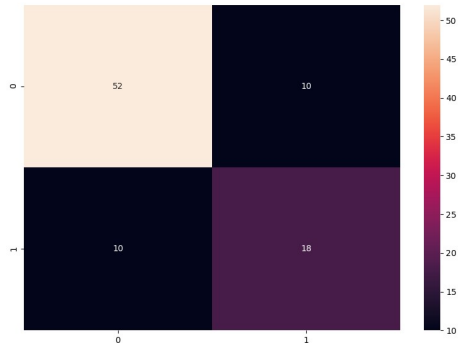
Classification report:

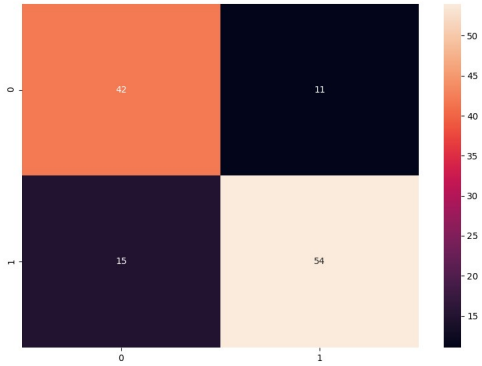
Pre-SMOTE

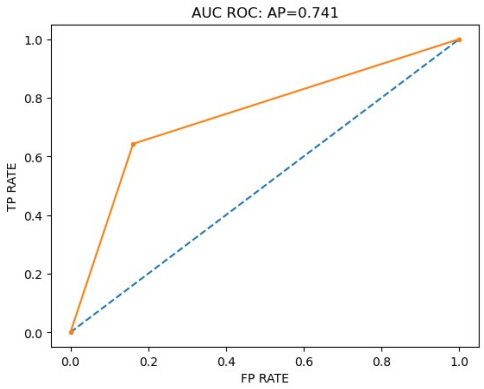
	precision	recall	f1-score	support
0	0.84	0.84	0.84	62
1	0.64	0.64	0.64	28
accuracy			0.78	90
macro avg	0.74	0.74	0.74	90
weighted avg	0.78	0.78	0.78	90

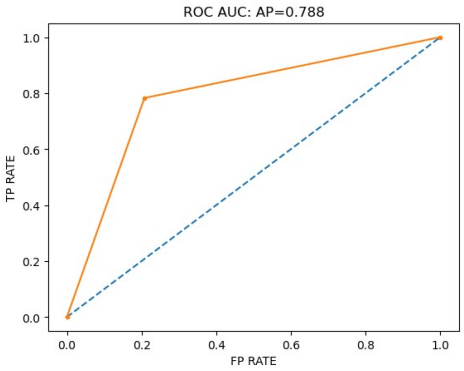
Post-SMOTE

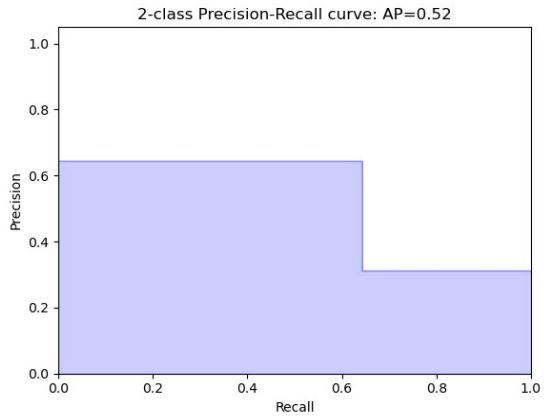
	precision	recall	f1-score	support
0	0.74	0.79	0.76	53
1	0.83	0.78	0.81	69
accuracy			0.79	122
macro avg	0.78	0.79	0.78	122
weighted avg	0.79	0.79	0.79	122

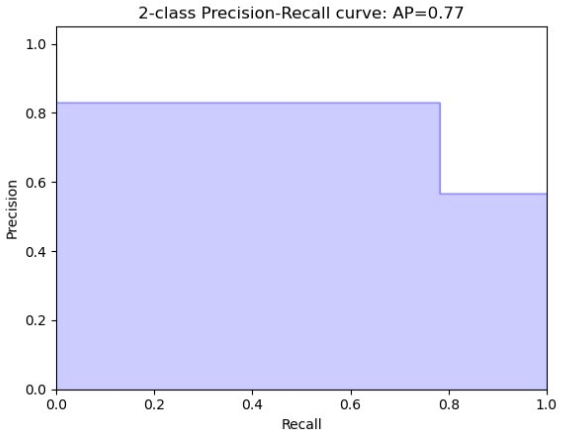












Attraverso l'algoritmo Decision Tree si può notare un netto miglioramento sotto tutti i punti di vista con l'implementazione di SMOTE, in particolare nella curva Precision-Recall.

Random Forest

Il Random Forest è una tecnica di apprendimento automatico utilizzata per risolvere problemi di regressione e classificazione. Utilizza una tecnica che combina più alberi di decisione per fornire soluzioni al problema per cui viene chiamato in causa. La predizione di ciascun albero può essere ottenuta o, attraverso la media delle predizioni di un albero per ogni esempio o, attraverso un meccanismo di votazione in cui tutti gli alberi votano la propria classificazione più probabile e l'esempio col maggior numero di voti sarà scelto come predizione finale.

Per mezzo di 20 alberi di decisone, i risultati ottenuti sono stati i seguenti:

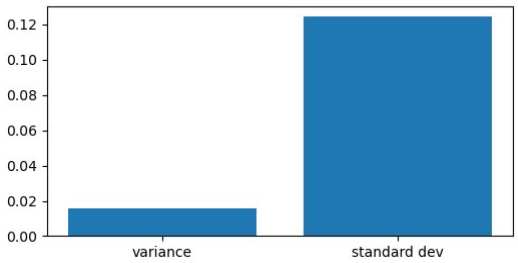
Cross Validation

Pre-SMOTE

cv_scores mean:0.7023728813559321

cv_score variance:0.01546696670816177

cv_score dev standard:0.12436626032876348

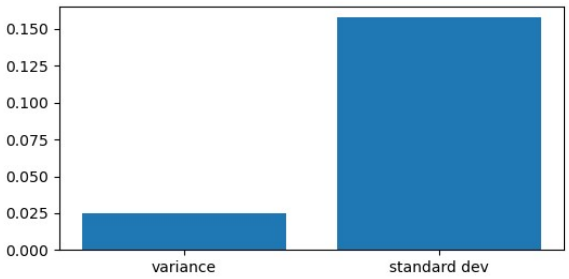


Post-SMOTE

cv_scores mean:0.7912978018669076

cv_score variance:0.024790554319273985

cv_score dev standard:0.1574501645577863



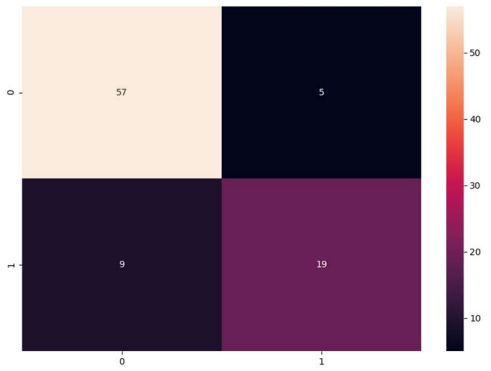
Classification report:

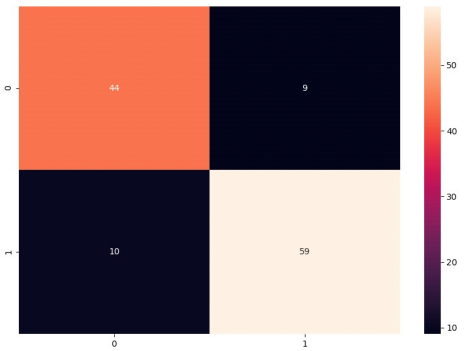
Pre-SMOTE

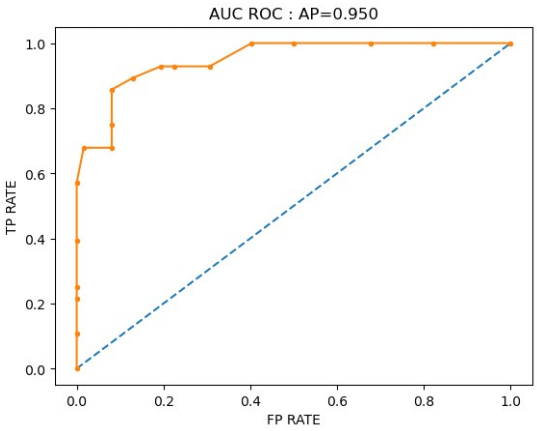
	precision	recall	f1-score	support
0	0.86	0.92	0.89	62
1	0.79	0.68	0.73	28
accuracy			0.84	90
macro avg	0.83	0.80	0.81	90
weighted avg	0.84	0.84	0.84	90

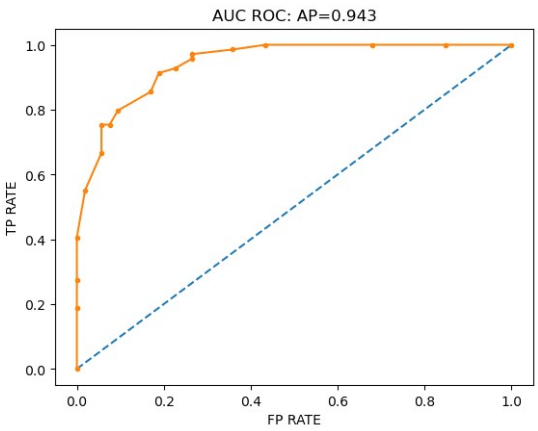
Post-SMOTE

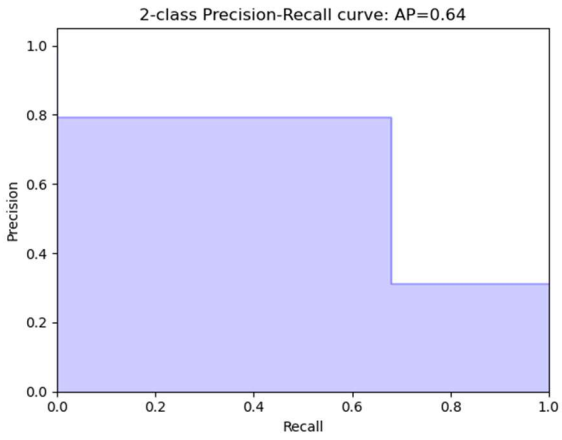
	precision	recall	f1-score	support
0	0.81	0.83	0.82	53
1	0.87	0.86	0.86	69
accuracy			0.84	122
macro avg	0.84	0.84	0.84	122
weighted avg	0.84	0.84	0.84	122

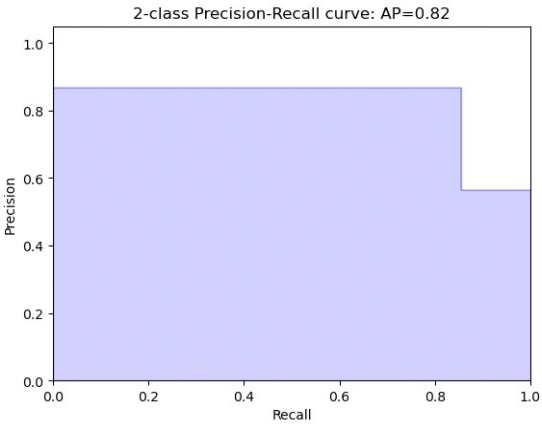












L’algoritmo Random Forest ha dimostrato risultati decisamente migliori sotto ogni punto di vista rispetto a tutti tra tutti gli algoritmi utilizzati. Con l’applicazione di SMOTE si ha una leggera variazione dell’area della curva ROC, mentre dei

peggioramenti dal punto di vista di cross validation. Si ha invece un miglioramento di F1 e AP.

Support Vector Machine (SVM)

Un modello SVM è una rappresentazione degli esempi come punti nello spazio, mappati in modo tale che gli esempi appartenenti alle due diverse categorie siano chiaramente separati da uno spazio il più possibile ampio. I nuovi esempi sono quindi mappati nello stesso spazio e la predizione della categoria alla quale appartengono viene fatta sulla base del lato nel quale ricade.

Gli algoritmi SVM utilizzano un insieme di funzioni matematiche definite come kernel. La funzione del kernel prende i dati come input e li trasformarli nella forma richiesta. Diversi algoritmi SVM utilizzano diversi tipi di funzioni del kernel. Queste funzioni possono essere di diversi tipi, come lineare, non lineare, polinomiale, funzione di base radiale (RBF) e sigmoide.

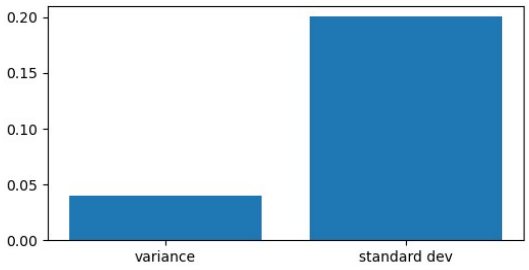
SVM(kernel='linear',gamma='auto',probability=True)

Cross Validation:

cv_scores mean:0.6989265536723164

cv_score variance:0.040151501803440911

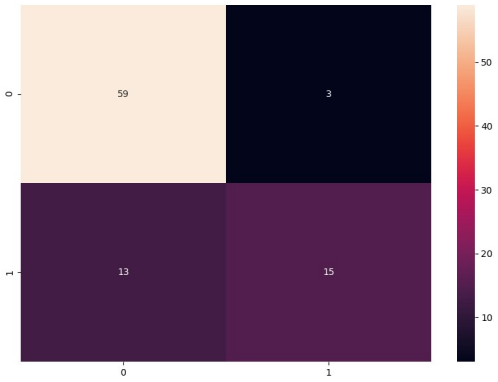
cv_score dev standard:0.20037839654873205

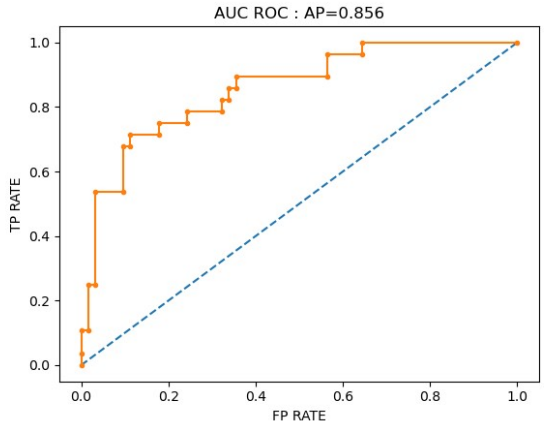


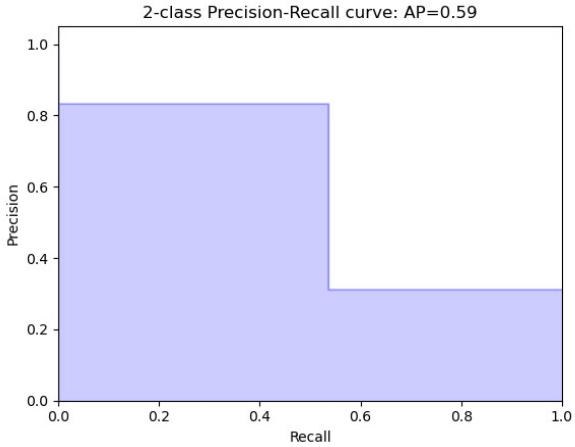
Classification report:

	precision	recall	f1-score	support
0	0.82	0.95	0.88	62
1	0.83	0.54	0.65	28

accuracy			0.82	90
macro avg	0.83	0.74	0.77	90
weighted avg	0.82	0.82	0.81	90







SVM(kernel=RBF,gamma='scale',probability=True)

Cross Validation:

Pre-SMOTE

cv_scores mean:0.6789265536723164

cv_score variance:4.189728366689038e-05

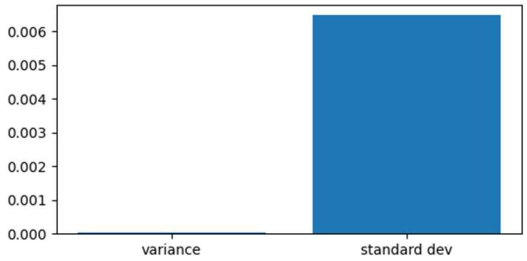
cv_score dev standard:0.006472811110088906

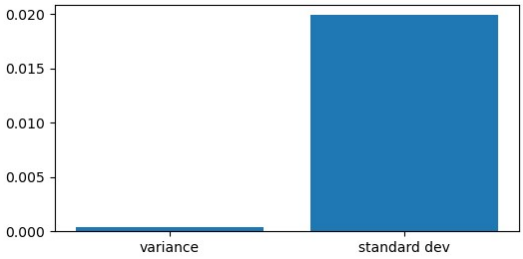
Post-SMOTE

cv_scores mean:0.5246311352002409

cv_score variance:0.00039629556187160866

cv_score dev standard:0.019907173628408645

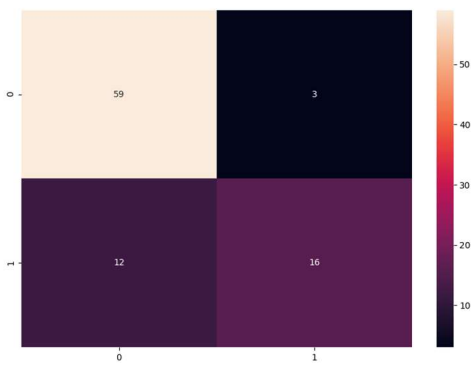


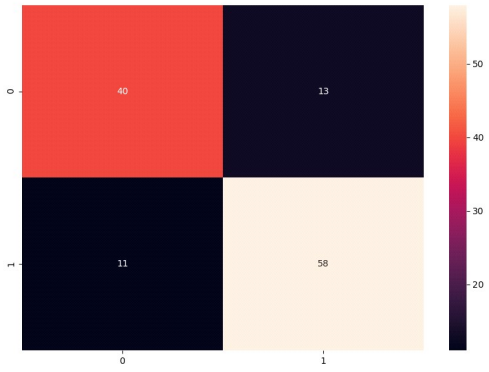


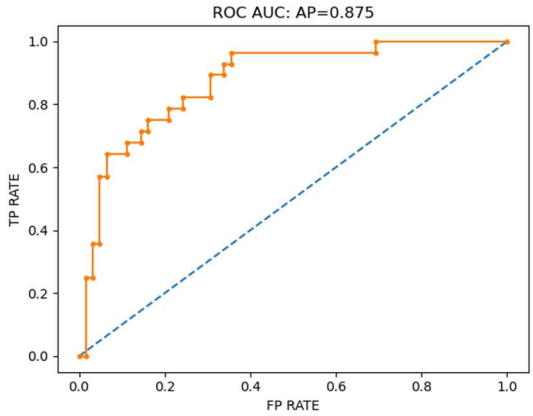
Classification report:

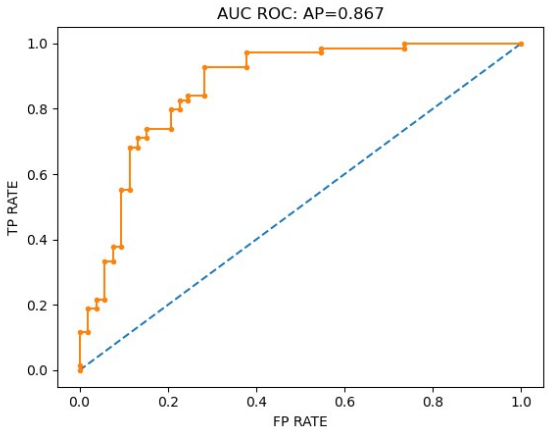
	precision	recall	f1-score	support
0	0.83	0.95	0.89	62
I	0.84	0.57	0.68	28
accuracy			0.83	90
macro avg	0.84	0.76	0.78	90
weighted avg	0.83	0.83	0.82	90

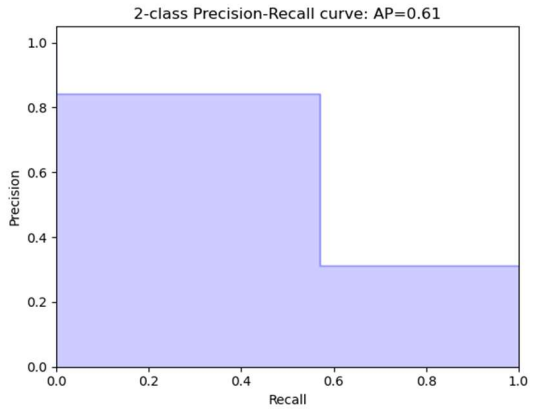
	precision	recall	f1-score	support
0	0.78	0.75	0.77	53
I	0.82	0.84	0.83	69
accuracy			0.80	122
macro avg	0.80	0.80	0.80	122
weighted avg	0.80	0.80	0.80	122

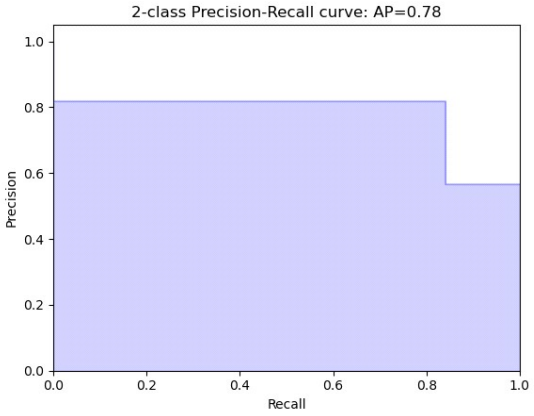












Gli algoritmi di SVM hanno riportato i migliori risultati dopo il Random Forest. Utilizzando RBF come kernel ci sono stati risultati migliori sotto ogni punto di vista, e applicando SMOTE gli unici miglioramenti si sono rilevati in F1 e Average Precision.

Multinomial Naive Bayes

Il Naive Bayes è un algoritmo classificatore basato sul teorema di Bayes. È una famiglia di classificatori statistici usati nel machine learning, nel quale le ipotesi di partenza sono molto semplificate. In particolar modo, si considerano indipendenti tra loro le varie caratteristiche (features) del modello.

Con un modello di eventi multinomiali, gli esempi (vettori di feature) rappresentano le frequenze con cui certi eventi sono stati generati da una distribuzione polinomiale (p1, …, pn) dove pi è la probabilità che l'evento i si verifichi.

Cross Validation:

Pre-SMOTE

cv_scores mean:0.6034025895814514

cv_score variance:0.0015796876627804709

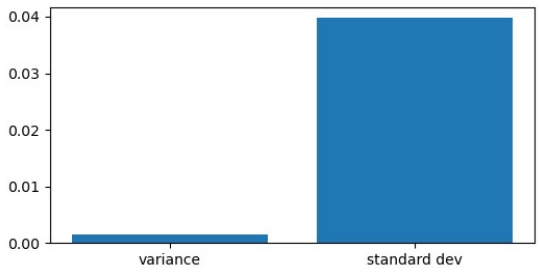
cv_score dev standard:0.03974528478675767

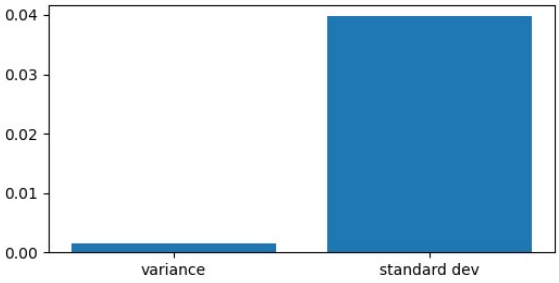
Post-SMOTE

cv_scores mean:0.6034025895814514

cv_score variance:0.0015796876627804709

cv_score dev standard:0.03974528478675767

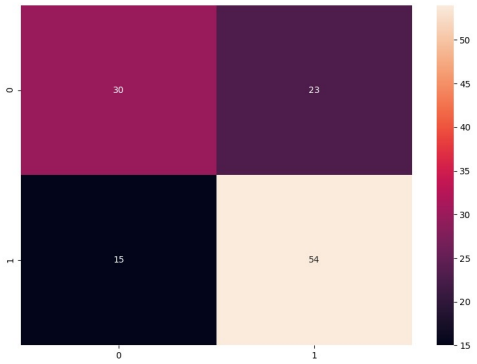


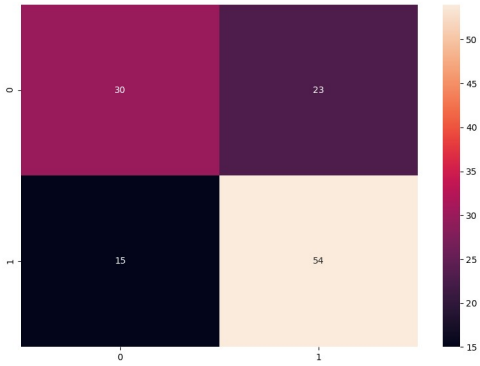


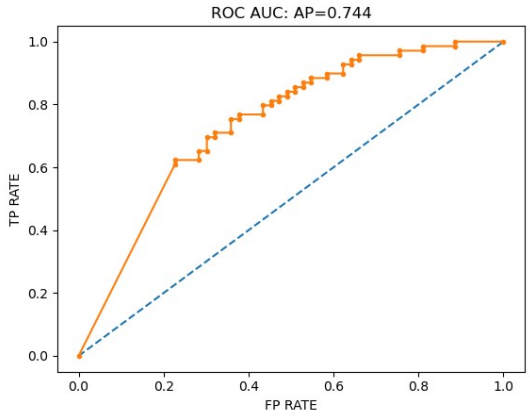
Classification report:

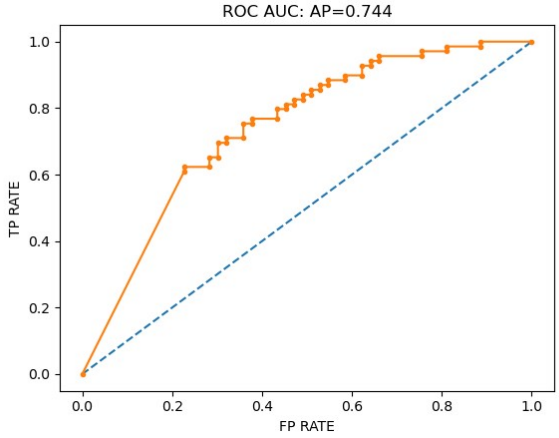
	precision	recall	f1-score	support
0	0.67	0.57	0.61	53
1	0.70	0.78	0.74	69
accuracy			0.69	122
macro avg	0.68	0.67	0.68	122
weighted avg	0.69	0.69	0.68	122

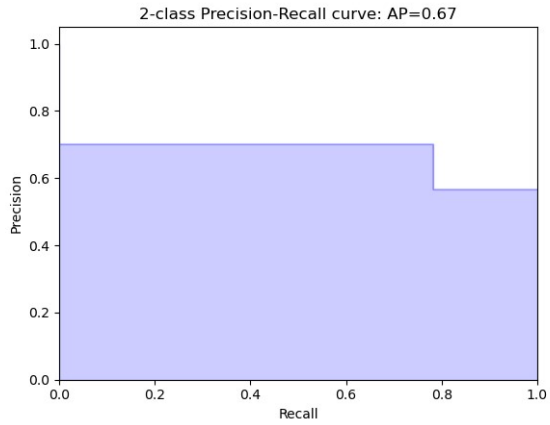
	precision	recall	f1-score	support
0	0.67	0.57	0.61	53
1	0.70	0.78	0.74	69
accuracy			0.69	122
macro avg	0.68	0.67	0.68	122
weighted avg	0.69	0.69	0.68	122

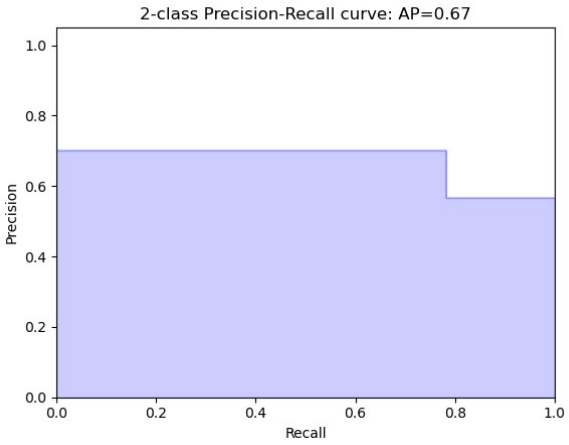












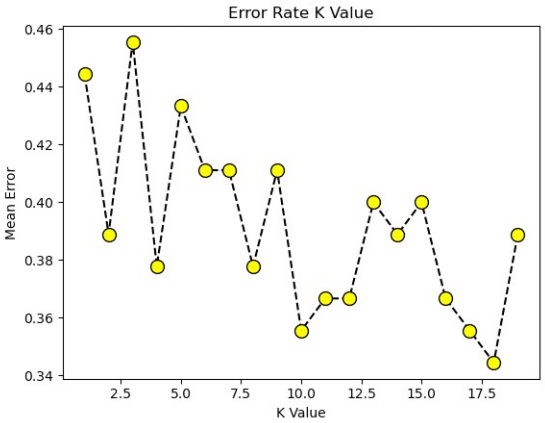
Attraverso SMOTE in Multinomial Naive Bayes si son verificati cambiamenti leggermenti vantaggiosi anche in questo caso nei confronti delle metriche F1 e Average Precision.

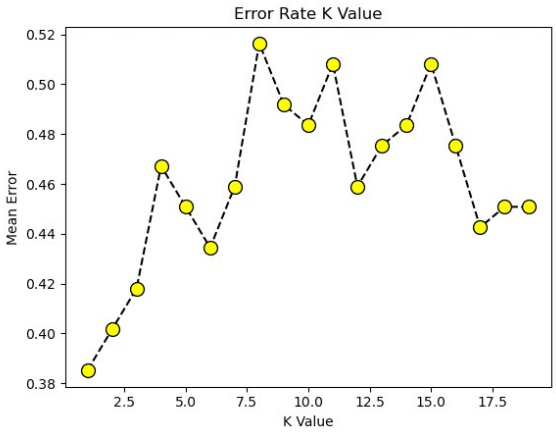
K-nearest neighbors

KNN è un algoritmo di apprendimento supervisionato, il cui scopo è quello di predire una nuova istanza conoscendo i data points che sono separati in diverse classi, individuando i k esempi più vicini a quella che si intende classificare.

Il grafico seguente, fornisce un suggerimento sulla scelta del numero di vicini per minimizzare l'errore medio. Si evince dal grafico che rispettivamente senza aver applicato SMOTE, con (numero di vicini) k = 10 e k=18, l'errore medio minimo commesso è di 0.35. Applicando SMOTE invece con k=2 si ha un errore medio minimo pari a 0.385.

Pre-SMOTE Post-SMOTE





Cross Validation:

cv_scores mean:0.6288700564971751

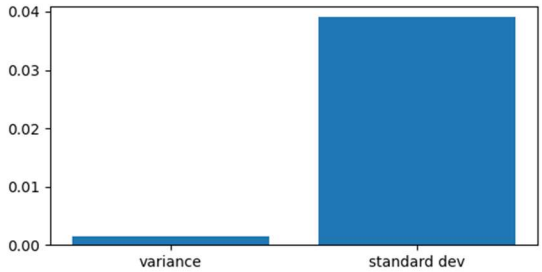
cv_score variance:0.0015222445657378152

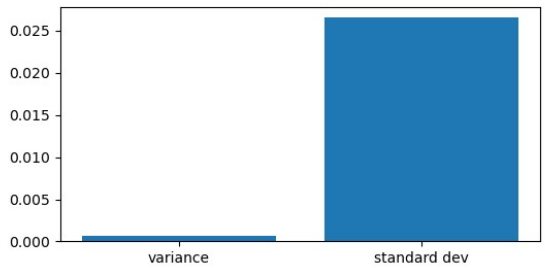
cv_score dev standard:0.03901595270831939

cv_scores mean:0.5887383318277627

cv_score variance:0.0007032850816313023

cv_score dev standard:0.026519522650894422

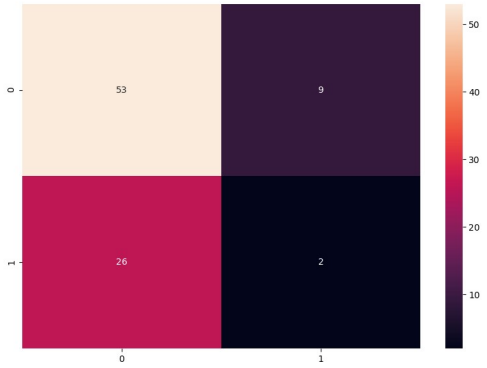


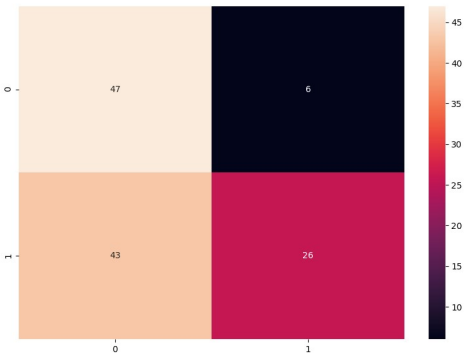


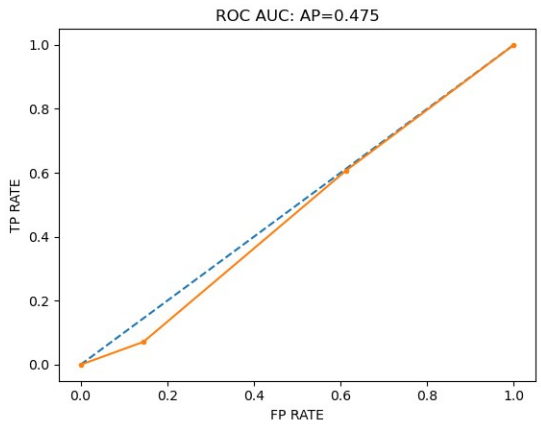
Classification report:

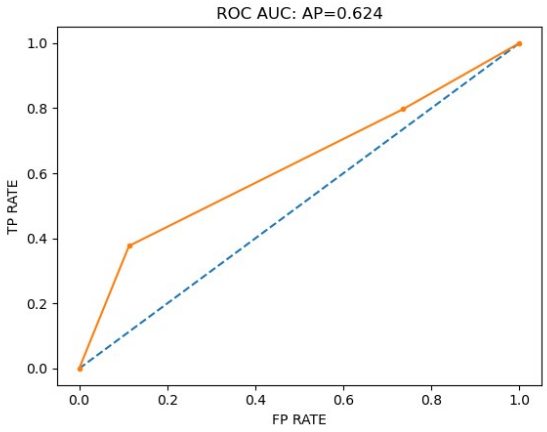
	precision	recall	f1-score	support
0	0.67	0.85	0.75	62
1	0.18	0.07	0.10	28
accuracy			0.61	90
macro avg	0.43	0.46	0.43	90
weighted avg	0.52	0.61	0.55	90

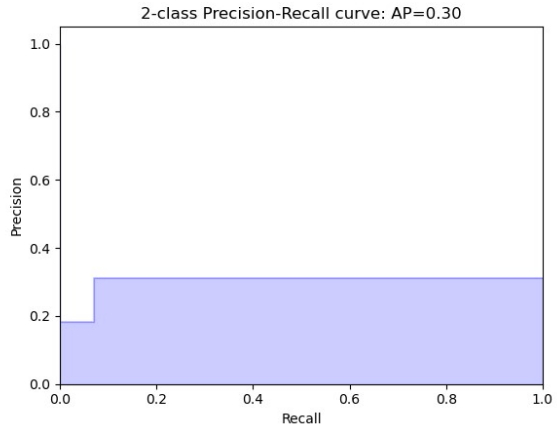
	precision	recall	f1-score	support
0	0.52	0.89	0.66	53
1	0.81	0.38	0.51	69
accuracy			0.60	122
macro avg	0.67	0.63	0.59	122
weighted avg	0.69	0.60	0.58	122

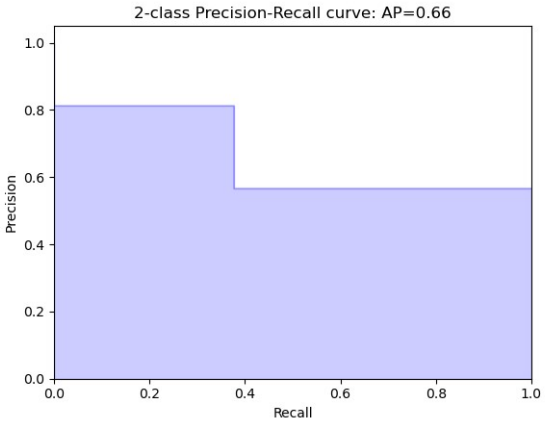












Utilizzando l'algoritmo K-nearest neighbors con k pari a 2, si possono notare netti miglioramenti, anche se pessimi in linea generale, dopo l'applicazione di SMOTE in particolare nell'Average Precision.

Logistic Regression(Features)

La regressione logistica è un algoritmo di apprendimento automatico che utilizza una funzione sigmoide e funziona al meglio su problemi di classificazione binaria,

sebbene possa essere utilizzato su problemi di classificazione multiclasse attraverso il metodo "uno contro tutti". È più comunemente usata quando i dati in questione hanno un output binario, quindi quando appartengono a una classe o a un'altra, o sono uno 0 o 1, nel caso in questione l'evento morte.

Cross Validation:

Creatinina sierica

cv_scores mean:0.6989265536723164

cv_score variance:0.00015150180344090163

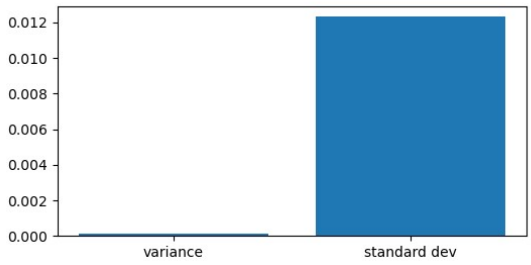
cv_score dev standard:0.01230860688465196

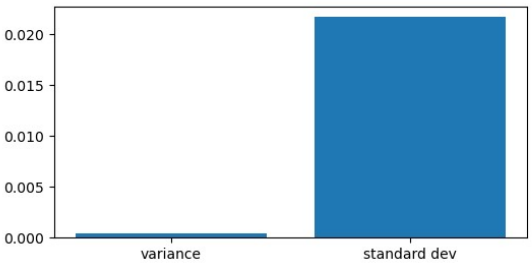
Frazione di eiezione

cv_scores mean:0.5862993074375188

cv_score variance:0.0033011609984947925

cv_score dev standard:0.057455730771567015





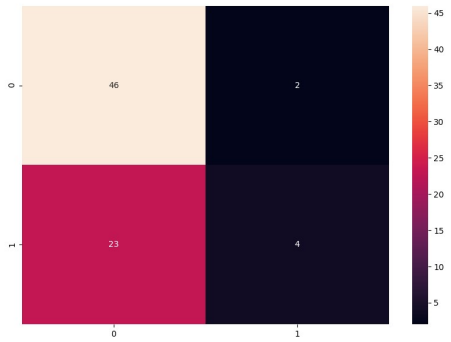
Classification Report:

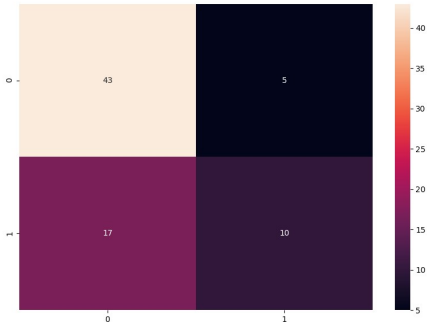
Creatinina sierica:

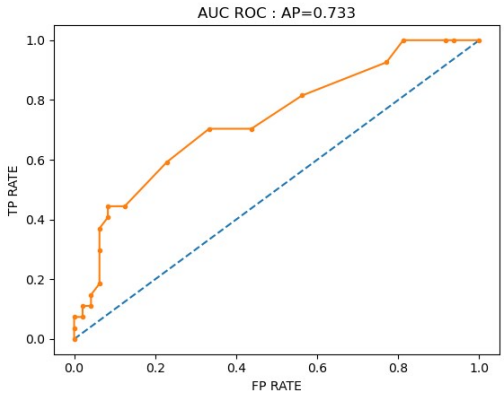
	precision	recall	f1-score	support
0	0.67	0.96	0.79	48
1	0.67	0.15	0.24	27
accuracy			0.67	75
macro avg	0.67	0.55	0.51	75
weighted avg	0.67	0.67	0.59	75

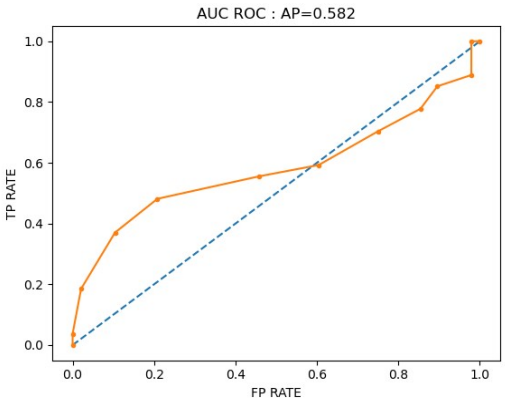
Frazione di eiezione

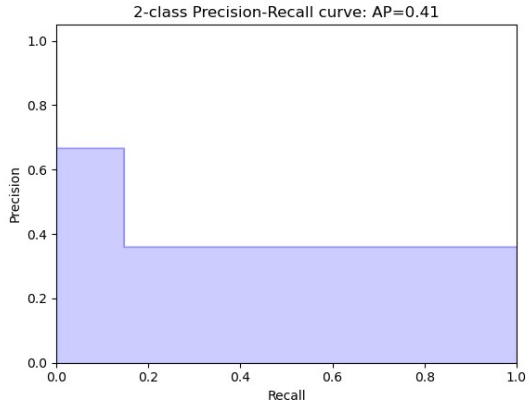
	precision	recall	f1-score	support
0	0.72	0.90	0.80	48
1	0.67	0.37	0.48	27
accuracy			0.71	75
macro avg	0.69	0.63	0.64	75
weighted avg	0.70	0.71	0.68	75

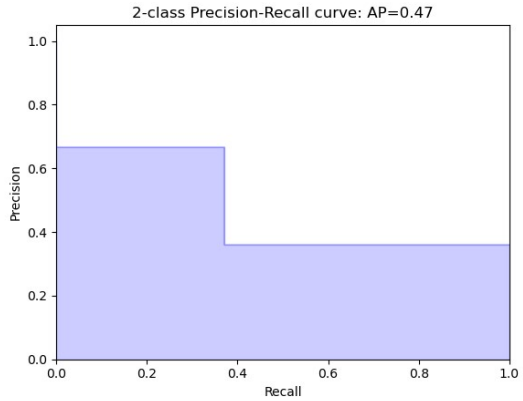

46
2
23
4
0
1
0
1
45
40
35
30
25
20
15
10
5


43
5
17
10
0
1
0
1
40
35
30
25
20
15
10
5


AUC ROC : AP=0.733
TP RATE
FP RATE
0.0
0.2
0.4
0.6
0.8
1.0


AUC ROC : AP=0.582
TP RATE
FP RATE
0.0
0.2
0.4
0.6
0.8
1.0


2-class Precision-Recall curve: AP=0.41
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0


2-class Precision-Recall curve: AP=0.47
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

Post-SMOTE

Cross Validation:

Creatinina sierica

cv_scores mean:0.6551641071966275

cv_score variance:0.0007954055116850142

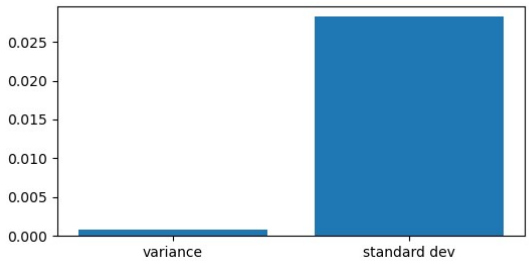
cv_score dev standard:0.028202934451666804

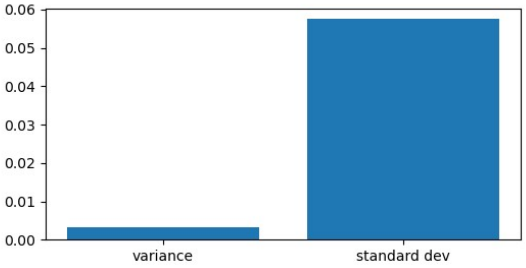
Frazione di eiezione

cv_scores mean:0.5862993074375188

cv_score variance:0.0033011609984947925

cv_score dev standard:0.057455730771567015





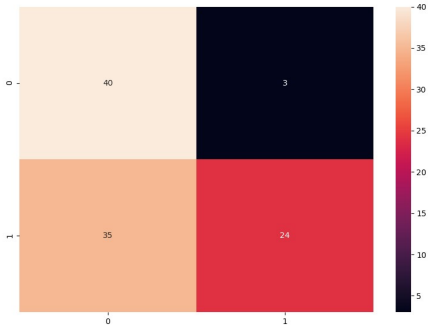
Classification Report:

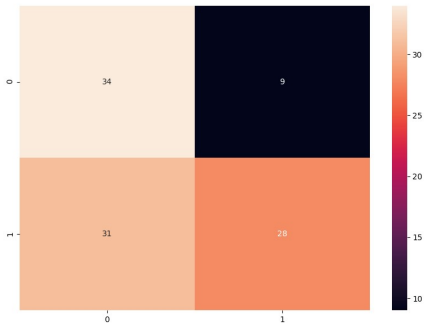
Creatinina sierica:

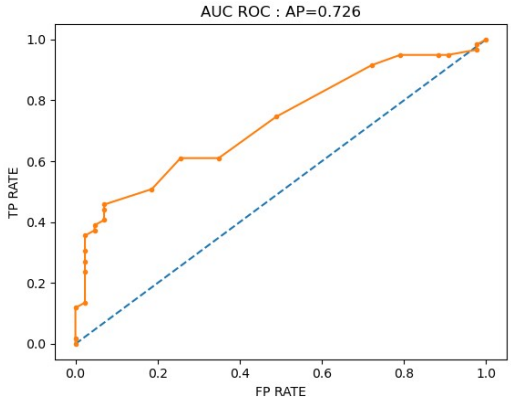
	precision	recall	f1-score	support
0	0.53	0.93	0.68	43
1	0.89	0.41	0.56	59
accuracy			0.63	102
macro avg	0.71	0.67	0.62	102
weighted avg	0.74	0.63	0.61	102

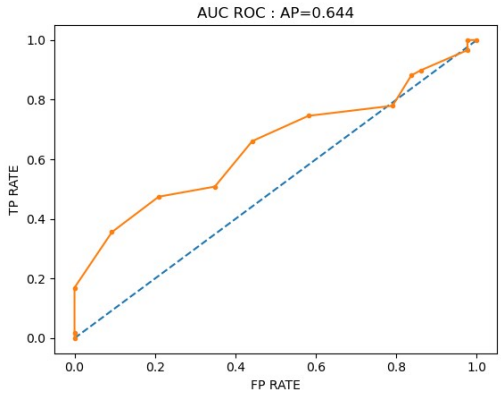
Frazione di eiezione

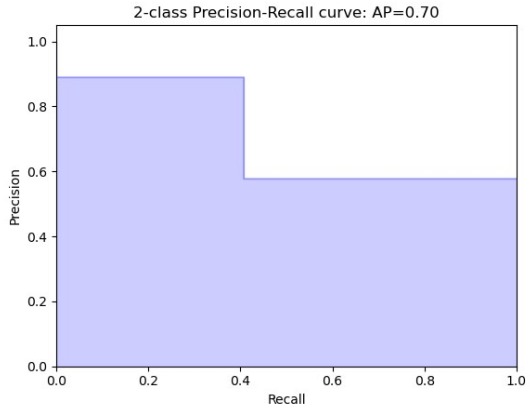
	precision	recall	f1-score	support
0	0.52	0.79	0.63	43
1	0.76	0.47	0.58	59
accuracy			0.61	102
macro avg	0.64	0.63	0.61	102
weighted avg	0.66	0.61	0.60	102

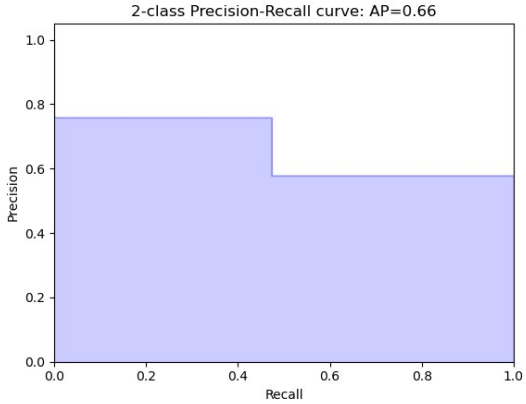

40
3
35
24
0
1
0
1


34
9
31
28
0
1
0
1


AUC ROC : AP=0.726
TP RATE
FP RATE
0.0
0.2
0.4
0.6
0.8
1.0


AUC ROC : AP=0.644
TP RATE
FP RATE
0.0
0.2
0.4
0.6
0.8
1.0

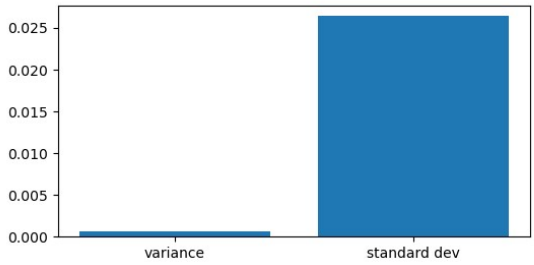

2-class Precision-Recall curve: AP=0.70
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

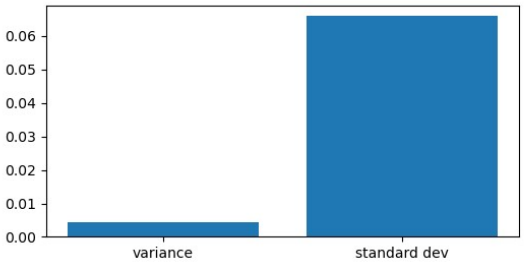

2-class Precision-Recall curve: AP=0.66
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

Random Forest (Features)

Random forest rispettivamente con le due feature, con SMOTE direttamente applicato

Creatinina sierica	Frazione di eiezione
Cross Validation:	
cv_scores mean:0.6058717253839205	cv_scores mean:0.6626618488407106
cv_score variance:0.0006971630625001752	cv_score variance:0.00434666622419846
cv_score dev standard:0.026403845600597182	cv_score dev standard:0.06592925165811045





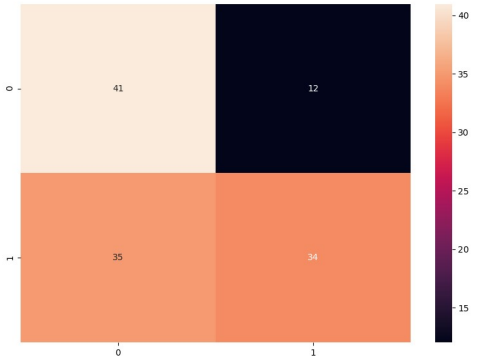
Classification Report:

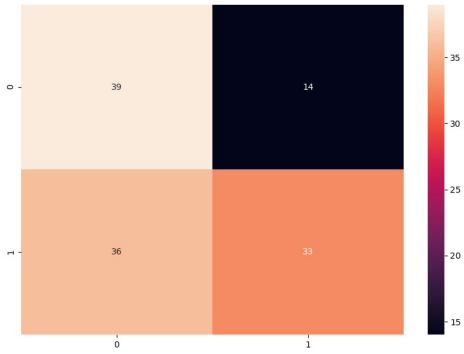
Creatinina sierica:

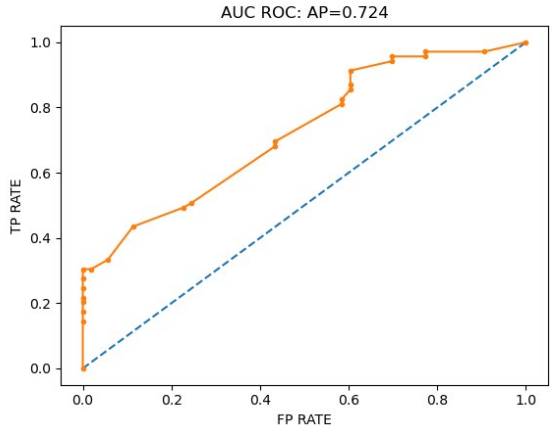
	precision	recall	f1-score	support
0	0.54	0.77	0.64	53
1	0.74	0.49	0.59	69
accuracy			0.61	122
macro avg	0.64	0.63	0.61	122
weighted avg	0.65	0.61	0.61	122

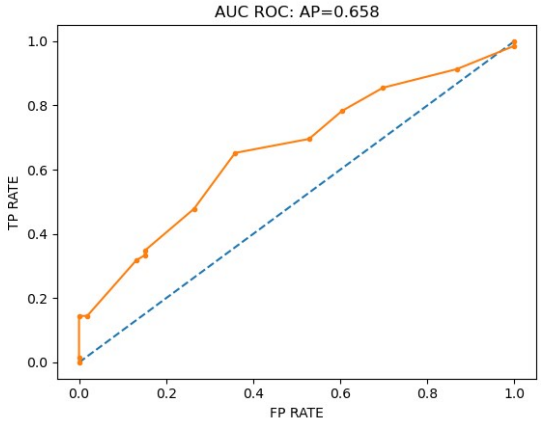
Frazione di eiezione

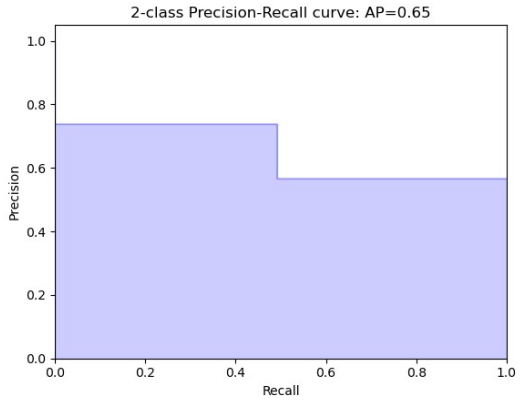
	precision	recall	f1-score	support
0	0.52	0.74	0.61	53
1	0.70	0.48	0.57	69
accuracy			0.59	122
macro avg	0.61	0.61	0.59	122
weighted avg	0.62	0.59	0.59	122

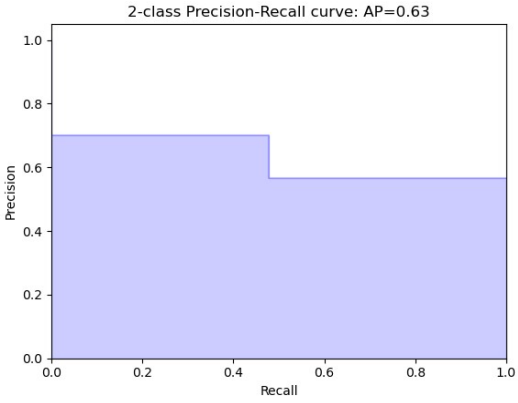

41
12
35
34
0
1
0
1
40
35
30
25
20
15


39
14
36
33
0
1
0
1
35
30
25
20
15


AUC ROC: AP=0.724
TP RATE
FP RATE
0.0
0.2
0.4
0.6
0.8
1.0


AUC ROC: AP=0.658
TP RATE
FP RATE
0.0
0.2
0.4
0.6
0.8
1.0


2-class Precision-Recall curve: AP=0.65
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0


2-class Precision-Recall curve: AP=0.63
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

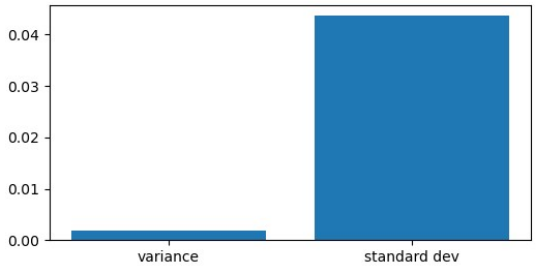
SVM(Features)

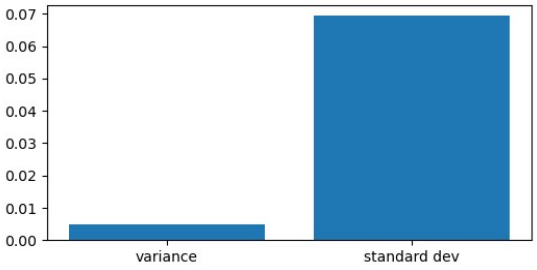
Support Vector Machine rispettivamente con le due feature, con SMOTE direttamente applicato

Creatinina sierica	Frazione di eiezione

Cross Validation:

cv_scores mean:0.6650406504065041	cv_scores mean:0.6357121348991267
cv_score variance:0.001900531514934072	cv_score variance:0.004803732255596837
cv_score dev standard:0.04359508590350605	cv_score dev standard:0.06930896230356387

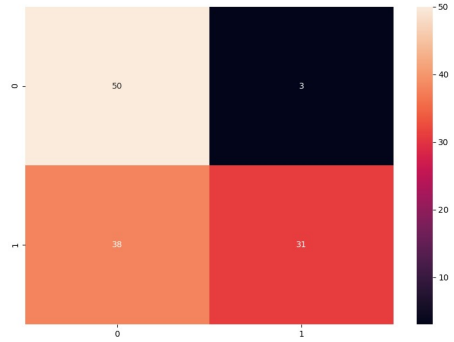


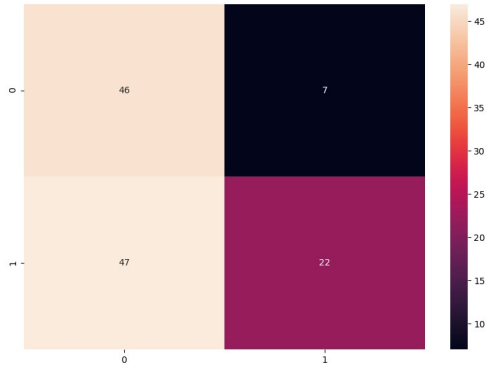


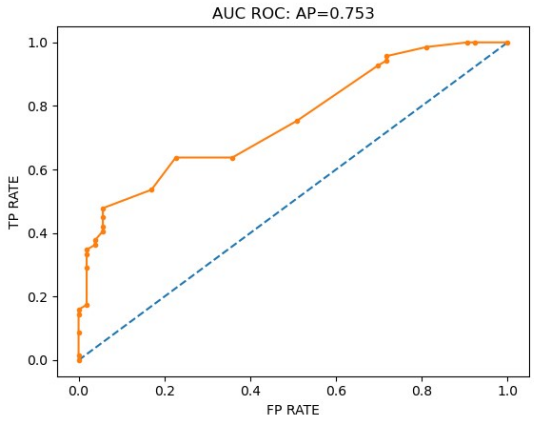
Classification report:

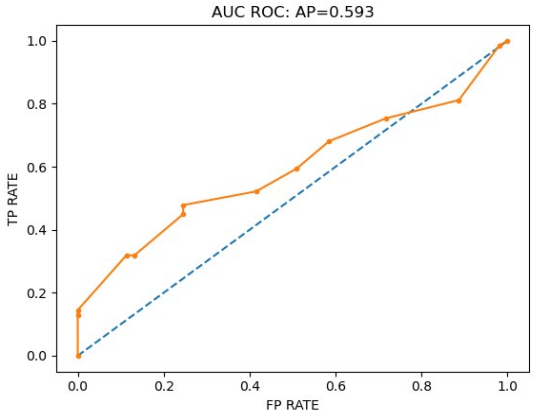
	precision	recall	fI-score	support
0	0.57	0.94	0.71	53
I	0.91	0.45	0.60	69
accuracy			0.66	122
macro avg	0.74	0.70	0.66	122
weighted avg	0.76	0.66	0.65	122

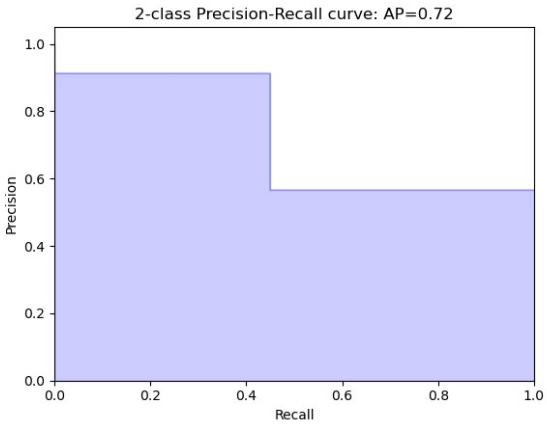
	precision	recall	fI-score	support
0	0.49	0.87	0.63	53
I	0.76	0.32	0.45	69
accuracy			0.56	122
macro avg	0.63	0.59	0.54	122
weighted avg	0.64	0.56	0.53	122

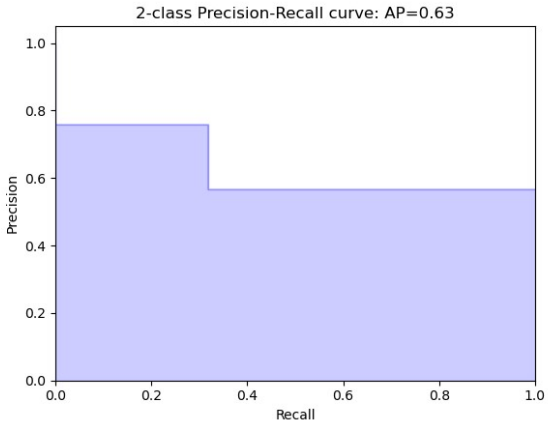












Attraverso l'algoritmo Logistic Regression, si notano migliori risultati sotto tutti i punti di vista, ad eccezione della curva ROC e dell'Average Precision, applicandolo sulla feature Frazione di eiezione.

Nonostante ciò, l'applicazione dei restanti due algoritmi (Random Forest e SVM con RBF kernel) sulla feature Creatinina sierica, ha prodotto valori migliori in particolare con l'algoritmo Support Vector Machine in ogni metrica.

K-Means

Il raggruppamento K-means è uno degli algoritmi più semplici ed efficaci appartenenti all'apprendimento non supervisionato.

Come menzionato nell'introduzione, gli algoritmi di apprendimento non supervisionato non richiedono l'inserimento di un valore etichettato, e dunque generalizzano un output senza l'intervento dell'utente.

Il K-Means identifica k centroidi, dei punti immaginari nello spazio che rappresentano il centro del raggruppamento, e colloca ogni punto al cluster più vicino. La variabile k definisce il numero di centroidi (o di cluster) da identificare, nel caso in questione k=2.

Ogni punto è assegnato ad un determinato cluster attraverso il calcolo dello scarto quadratico medio, la distanza tra il punto e il centroide. Per ogni punto, l'errore è la distanza dal centroide del cluster a cui esso è assegnato.

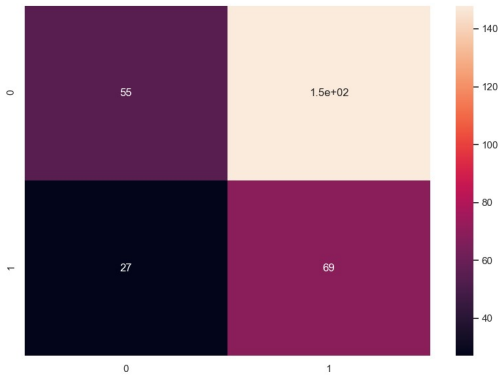
Classification report:

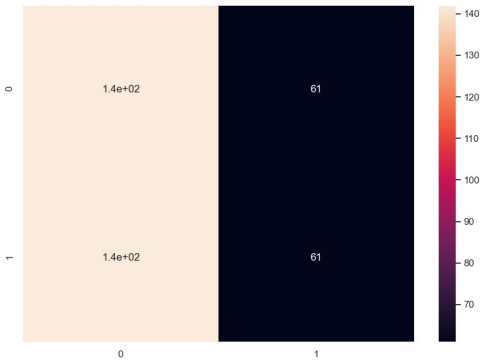
Pre-SMOTE

	precision	recall	f1-score	support
0	0.67	0.27	0.39	203
1	0.32	0.72	0.44	96
accuracy			0.41	299
macro avg	0.49	0.49	0.41	299
weighted avg	0.56	0.41	0.40	299

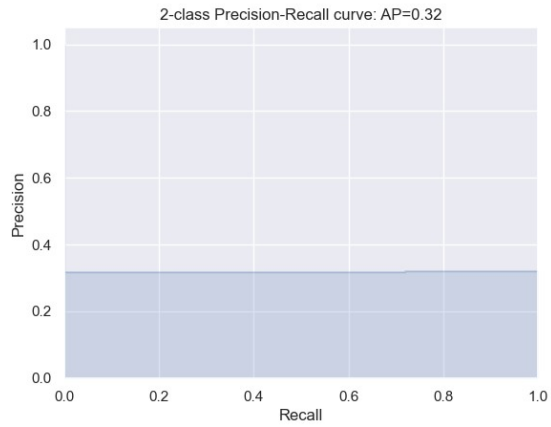
Post-SMOTE

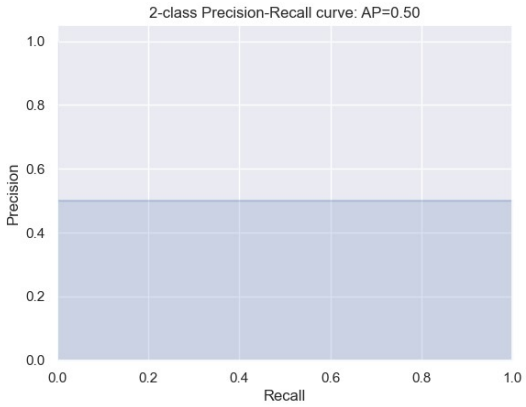
	precision	recall	f1-score	support
0	0.50	0.70	0.58	203
1	0.50	0.30	0.38	203
accuracy			0.50	406
macro avg	0.50	0.50	0.48	406
weighted avg	0.50	0.50	0.48	406





L’algoritmo K-Means, unico del tipo di apprendimento non supervisionato, assieme al KNN ha prodotto i peggiori risultati tra tutti gli algoritmi applicati.





Modelli a confronto

Apprendimento supervisionato

Algoritmo	Accuratezza	Varianza	Deviaz. Standard	F1	Average Precision	AUC ROC
Decision Tree	0.78	0.03	0.17	0.74	0.52	0.741
Decision Tree(SMOTE)	0.79	0.022	0.15	0.79	0.77	0.788
Random Forest	0.84	0.015	0.124	0.81	0.64	0.95
Random Forest(SMOTE)	0.84	0.025	0.16	0.84	0.82	0.943
SVM (linear kernel)	0.82	0.04	0.2	0.77	0.59	0.856
SVM (RBF kernel)	0.83	4.18 e-05	0.006	0.78	0.61	0.875
SVM(RBF kernel SMOTE)	0.80	0.0004	0.02	0.80	0.78	0.867
Multinomial Naive Bayes	0.67	0.002	0.05	0.58	0.36	0.781
Multinomial Naive Bayes(SMOTE)	0.69	0.002	0.04	0.68	0.67	0.744
KNN	0.61	0.002	0.04	0.43	0.3	0.475
KNN(SMOTE)	0.60	0.0007	0.03	0.59	0.66	0.624

Algoritmo	Feature	Accuratezza	Varianza	Deviaz. Standard	F1	Average Precision	AUC ROC
Logistic Regression	Creatinina sierica	0.67	0.0001	0.012	0.51	0.41	0.733
Logistic Regression(SMOTE)	Creatinina sierica	0.63	0.0007	0.028	0.63	0.70	0.726
Logistic Regression	Frazione di eiezione	0.71	0.0004	0.021	0.71	0.47	0.528
Logistic Regression(SMOTE)	Frazione di eiezione	0.61	0.003	0.057	0.61	0.66	0.644
Random Forest(SM OTE)	Creatinina sierica	0.61	0.0007	0.026	0.61	0.65	0.724
Random Forest(SM OTE)	Frazione di eiezione	0.59	0.004	0.065	0.59	0.63	0.658
SVM (RBF kernel, SMOTE)	Creatinina sierica	0.66	0.002	0.043	0.66	0.72	0.753
SVM (RBF kernel, SMOTE)	Frazione di eiezione	0.56	0.005	0.07	0.54	0.63	0.593

Apprendimento non supervisionato

Algoritmo	Accuratezza	F1	Average Precision
K-means	0.41	0.41	0.32
K-means(SMOTE)	0.50	0.48	0.50

CONCLUSIONI

Il miglior algoritmo per la classificazione binaria del dataset proposto, risulta essere il Random Forest, seguito da SVM. La feature più rilevante ritrovata è stata la creatinina sierica, che assieme alla seconda feature più rilevante, la frazione di eiezione, ha prodotto risultati discreti quando utilizzata singolarmente per effettuare la classificazione.

L'apprendimento non supervisionato, anche se rappresentato da un solo algoritmo, non si è dimostrato molto accurato per la classificazione.

L'applicazione di SMOTE ha dimostrato in generale un miglioramento delle performance quando applicato, anche se ha portato quasi sempre a "falsare" i valori di F1 e average precision perché avveniva un aumento di istanze della classe in minoranza.