Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Studio sulla classificazione di dati radiomici relativi allo stato tumorale in pazienti affetti da carcinoma della prostata, per mezzo di variational autoencoder

Y. Galluzzo, A. Fiannaca, M. La Rosa, L. La Paglia, A. Urso

Rapporto Tecnico N.:
RT-ICAR-PA-20-02 **dicembre 2020**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 8-9C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sede di Palermo, Via Ugo La Malfa 153, 90146 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Studio sulla classificazione di dati radiomici relativi allo stato tumorale in pazienti affetti da carcinoma della prostata, per mezzo di variational autoencoder

Y. Galluzzo [1], A. Fiannaca [2], M. La Rosa [2], L. La Paglia [2], A. Urso [2]

Rapporto Tecnico N.:
RT-ICAR-PA-20-02

Data:
dicembre 2020

---

[1] Università degli Studi di Palermo, Dipartimento di Matematica e Informatica, Via Archirafi 34, 90123, Palermo.
[2] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Palermo, Via Ugo La Malfa 153, 90146, Palermo.

1. Introduzione

La classificazione nel machine learning è una forma di analisi dei dati, che coinvolge lo sviluppo di modelli d'apprendimento, capaci di fare previsioni basandosi sui dati; in altre parole, la classificazione predice le etichette (*target*) di categorie che verranno applicate ai dati. Nell'ambito della bioinformatica applicata alla medicina di precisione, ad esempio, la classificazione può suggerire se un paziente è malato o è sano, oppure se ha un tipo di malattia piuttosto che un altro.

La classificazione è un problema molto complesso sia per la difficoltà intrinseca della stessa, che per la quantità di dati che bisogna analizzare per la creazione dei modelli. Negli anni sono stati sviluppati vari metodi di classificazione, la cui applicazione ha offerto interessanti risultati nell'ambito della bioinformatica. In questo studio verranno testati quelli più adatti per la tipologia di dato che è oggetto di studio.

2. Dataset radiomico

Nel campo della medicina, la radiomica è un metodo che estrae un gran numero di caratteristiche dalle immagini mediche radiografiche utilizzando algoritmi di caratterizzazione dei dati. Queste caratteristiche, chiamate *caratteristiche radiomiche*, informazioni di tipo quantitativo, non sono rilevabili semplicemente tramite la vista dell'operatore sanitario.

Negli ultimi anni il recente studio delle caratteristiche radiomiche in campo oncologico ha portato a risultati interessanti: ad esempio, si è concluso che le caratteristiche radiomiche possono essere utili per identificare i pazienti ad alto rischio di sviluppare metastasi a distanza, per quando riguarda il cancro ai polmoni [1], guidando cosi i medici a selezionare il trattamento efficace e puntuale per i singoli pazienti.
Studi radiomici hanno dimostrato che i marcatori basati sull'immagine hanno il potenziale di fornire informazioni ortogonali alla stadiazione e ai biomarcatori e di migliorare la prognosi.

La radiomica, in sintesi, nasce per sviluppare strumenti di supporto decisionale, e dunque ai dati ricavati vengono applicate metodiche di Machine Learning, al fine di costruire un modello predittivo clinicamente rilevante, grazie all'uso delle *caratteristiche radiomiche* ricavate dalle immagini mediche (TC, PET, SPECT etc.).
Il dataset utilizzato, in questi esperimenti, è stato ricavato dalle scansioni

Contrast	Entropy	...	PSA	ETA	Tipologia	Status
6032	0	...	2.2	71	0	0
6426	-2.7726	...	1.4	57	1	0
16946	-38.66	...	2.97	75	2	1

Table 1: Prototipo Dataset Radiomico utilizzato

di imaging PET / TC 18F-Cho, che sono state eseguite presso l'Unità di Medicina Nucleare dell'Istituto G.Giglio di Cefalù (Italia) [2].

A fine esemplificativo viene riportato in Table 1, il prototipo del dataset utilizzato.

Nel dettaglio, il dataset utilizzato, è formato da *45 samples* e *108 caratteristiche radiomiche* (dimensione del tumore, contrasto, entropia, ecc). Le caratteristiche riportate per ogni sample del dataset sono in totale *112*; Infatti, oltre alle caratteristiche radiomiche per ogni PET nel dataset troviamo associati i valori riguardanti, ad esempio, PSA, ETA (valori clinici), posizione della lesione (T=1, N=0 o M=2), *"Status"* del tumore: benigno (0), maligno (1). Volendo utilizzare un approccio basato sul machine learning, il dataset presenta sicuramente una criticità legata al numero esiguo di campioni presenti nel dataset oggetto di studio. Questa problematica, viene affrontata dagli esperimenti svolti. Infatti per porre una soluzione, momentanea ma estremamente utile, a questa criticità si cerca di applicare un buon metodo di Data Augmentation, al fine di utilizzare al meglio i modelli d'apprendimento.

3. Pipeline dello studio

In questo studio è stata realizzata un pipeline composta da 4 fasi:

1. **normalizzazione dei valori delle features** (necessaria in quanto ogni feature è scalata su un proprio range);
2. **riduzione del numero di caratteristiche tramite proiezione dello spazio delle features** (dato il rapporto tra samples e numero di features, è necessario ridurre le stesse per non incorrere in problematiche, quali la *curse of dimensionality*);
3. **apprendimento del modello di classificazione** (verranno testati più modelli di machine learning, che utilizzano approcci computazionali differenti);

4. **validazione dei risultati** (verranno comparati i risultati dell'addestramento per individuare il modello più indicato per il problema).

3.1. Applicazione modelli di ML per la classificazione

Inizialmente sono stati effettuati esperimenti utilizzando alcuni modelli standard di ML sul dataset (Figure 1, 3), al fine di classificare i *sample* in base al *target* STATUS del tumore (0=benigno o 1 =maligno). Sono stati utilizzati i modelli d'apprendimento: Naive Bayes (NB), Support Vector Machines (SVM), Random Forest (RF), Logistic Regression (Logit).
In particolare, i seguenti esperimenti sono stati effettuati usando una 5-fold cross validation applicando diversi metodi di standardizzazione dei dati. Gli esperimenti sono stati svolti, facendo inizialmente un pre-processing sui dati, necessario in quanto ogni caratteristica presenta intervalli di valori diversi, standardizzando con le tecniche:

- **Standard Scaler**, trasforma i dati in modo tale che la loro distribuzione avrà un valore medio uguale a 0 e deviazione standard uguale a 1. Data la distribuzione dei dati, ogni valore nel set di dati avrà il valore medio sottratto, e poi diviso per la deviazione standard dell'intero set di dati (o caratteristica nel caso di dati multivariati).

- **MinMax Scaler**, trasforma le caratteristiche scalando ognuna di esse in un determinato intervallo. Questo stimatore, scala e traduce ogni caratteristica individualmente in modo tale che si trovi in un particolare intervallo, ad esempio tra zero e uno (default).

I risultati iniziali confermano che, come anticipato, i modelli non riescono ad apprendere in modo sufficiente sul dataset (troppo esiguo).

A seguito dei primi test, si è provato ad utilizzare, per espandere il dataset "critico" (45 sample), una tecnica di sovraccampionamento.

Si tratta di un tipo di incremento dei dati per la classe di minoranza e si chiama *Tecnica di Sovraccampionamento Sintetico delle Minoranze*, o SMOTE in breve. Inizialmente si è applicata la tecnica su tutti e 45 i samples, arrivando dunque ad avere come dataset un numero di samples pari a 72.

Di seguito vengono presentati gli esperimenti svolti sul dataset sovraccampionato (72 samples):

Risultati metriche
(***senza*** *oversampling*):

Accuracy SVM: 0.51
Accuracy RF: 0.6
Accuracy NB: 0.48
Accuracy LOGIT: 0.46

Precision SVM: 0.5
Precision RF: 0.58
Precision NB: 0.48
Precision LOGIT: 0.45

Recall SVM: 0.5
Recall RF: 0.63
Recall NB: 0.54
Recall LOGIT: 0.45

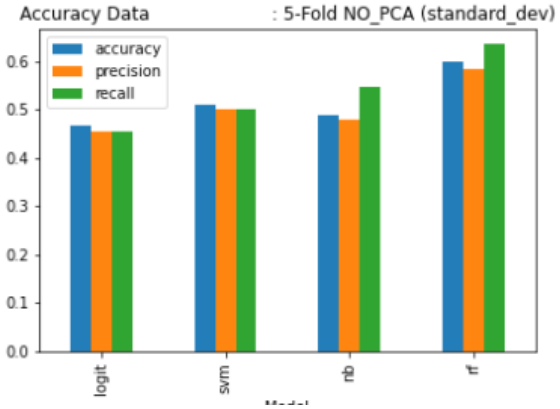


Figure 1: Risultati esperimenti con normalizzazione StandardScaler e senza PCA.

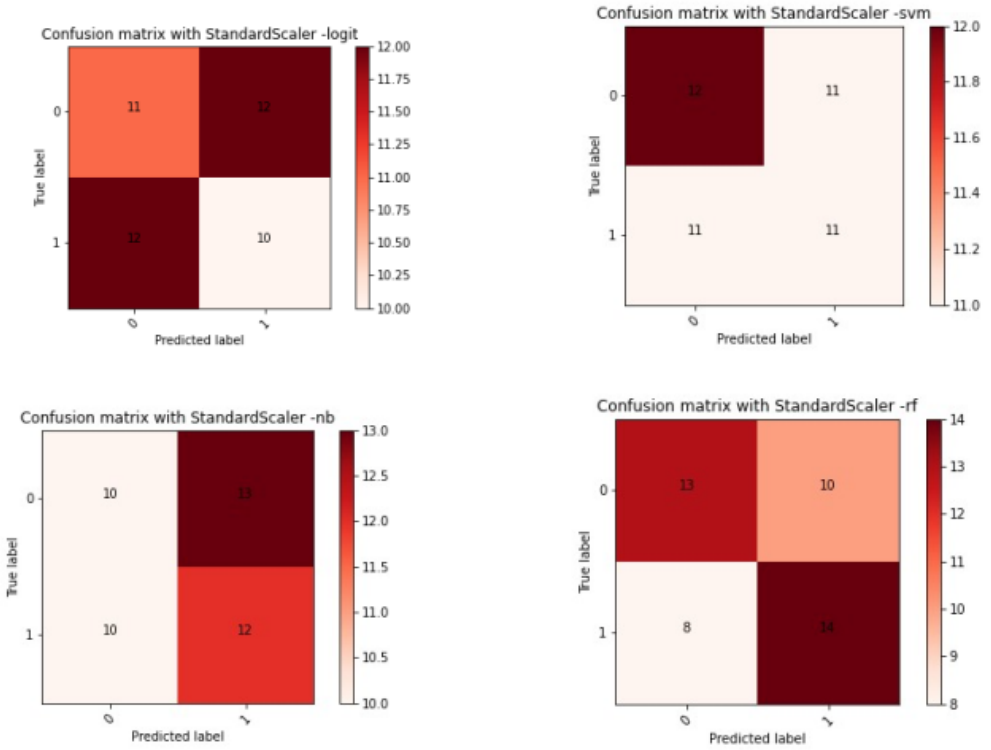


Figure 2: Matrici di confusione: esperimenti con normalizzazione StandardScaler e senza PCA.

Risultati metriche
***(senza** oversampling):*

Accuracy SVM: 0.4888888888888889
Accuracy RF: 0.6
Accuracy NB: 0.4888888888888889
Accuracy LOGIT: 0.4888888888888889

Precision SVM: 0.47619047619047616
Precision RF: 0.5833333333333334
Precision NB: 0.48
Precision LOGIT: 0.4782608695652174

Recall SVM: 0.45454545454545453
Recall RF: 0.6363636363636364
Recall NB: 0.5454545454545454
Recall LOGIT: 0.5

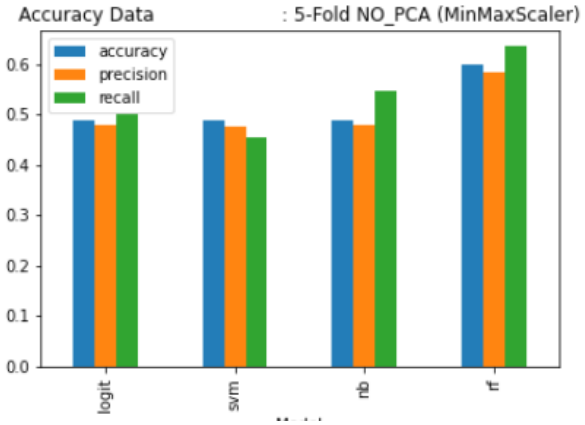


Figure 3: Risultati esperimenti con normalizzazione MinMaxScaler e senza PCA.

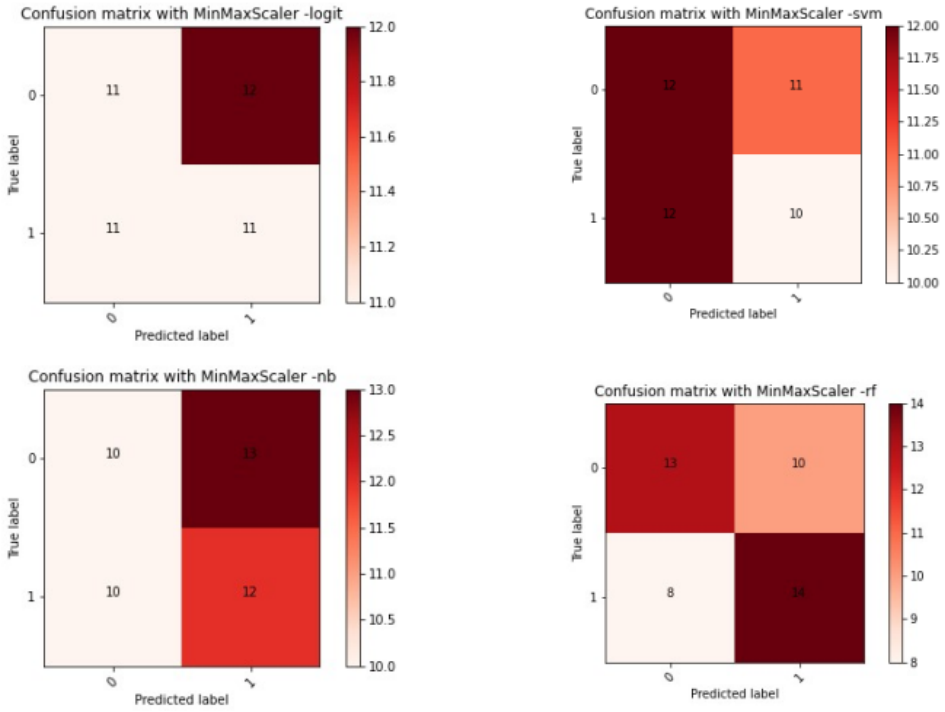
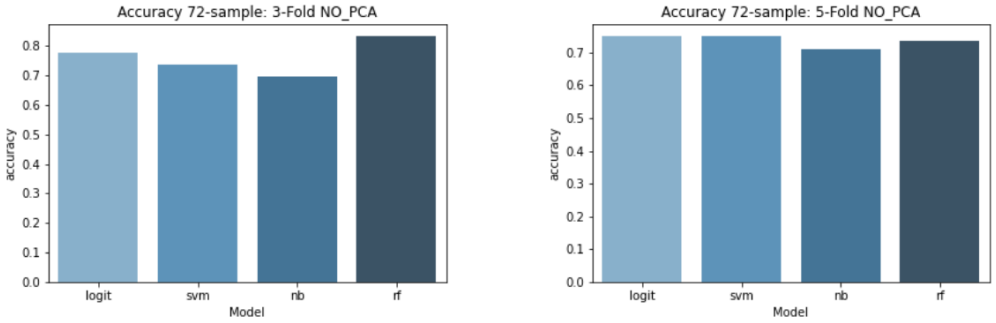


Figure 4: Matrici di confusione: esperimenti con normalizzazione MinMaxScaler e senza PCA.

- Senza uso di riduzione di feature (non viene usata la PCA), i test eseguiti sono stati svolti su 3-fold e su 5-fold, utilizzando come modelli di apprendimento Naive Bayes, Logistic Regression (logit), Support Vector Machine (SVM con kernel = rbf), e Random Forest (RF). Per la standardizzazione dei valori dei dati è stato usato solo il MinMaxScaler.



[0.77, 0.73, 0.69, 0.83]

[0.75, 0.75, 0.70, 0.73]

Figure 5: Risultati esperimenti con normalizzazione MinMaxScaler e senza PCA.

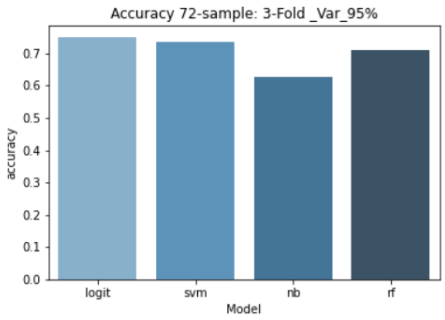
- Con l'uso di PCA per la riduzione delle 108 features del dataset, i test eseguiti sono stati svolti su 3-fold e su 5-fold, utilizzando come modelli di apprendimento Naive Bayes, Logistic Regression (logit), Support Vector Machine (SVM con kernel = rbf), e Random Forest (RF).Per la standardizzazione dei valori dei dati è stato usato solo il MinMaxScaler.

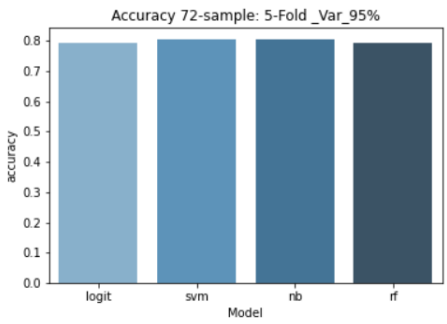
In Figure 11, 12, 13, 14, 15, invece, si riportano i risultati in **media** dell'accuracy utilizzando i modelli citati sopra. L'esperimento è stato **ripetuto 10 volte**.

Notiamo che il modello che sembra essere più stabile è il Random Forest, che mantiene sempre una percentuale d'accuratezza superiore al 75% (Figure 16).

Viene rappresentato in Figure 17 la tendenza dei modelli (NB, RF, Logit e SVM) dell'accuracy in media su 10 esecuzioni, senza la riduzione delle features (senza PCA).

Anche in questo caso, il modello Random Forest risulta essere quello con predizione più "accurata".

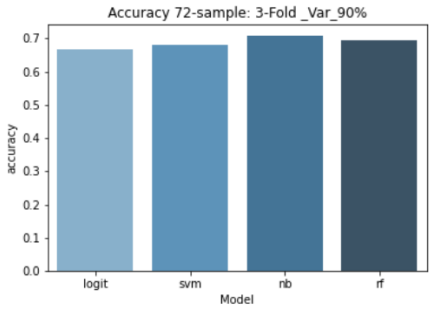


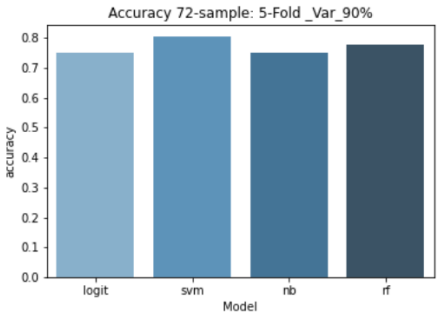



```
[0.75, 0.73, 0.625, 0.70]
```

```
[0.79, 0.80, 0.80, 0.79]
```

Figure 6: Risultati esperimenti con normalizzazione MinMaxScaler, mantenendo varianza a 95% (29 caratteristiche).

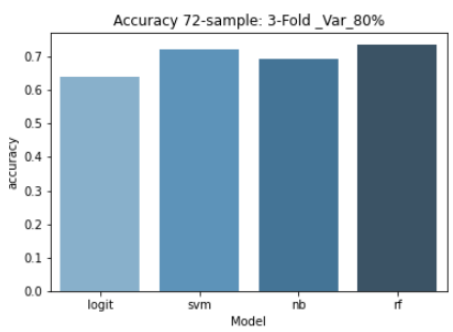




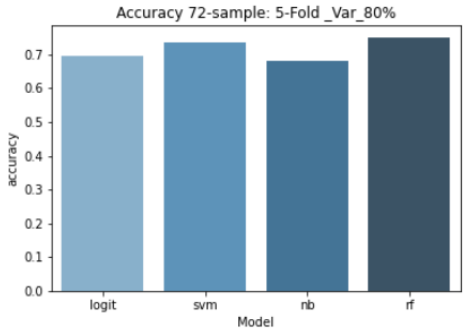

```
[0.66, 0.68, 0.70 0.69]
```

```
[0.75, 0.80, 0.75, 0.77]
```

Figure 7: Risultati esperimenti con normalizzazione MinMaxScaler, mantenendo varianza a 90% (23 caratteristiche).

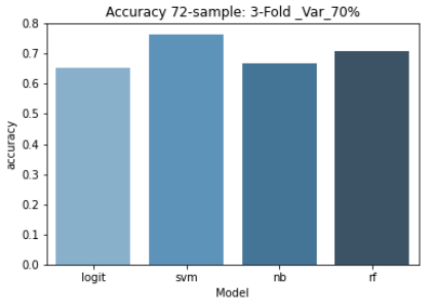


[0.63 0.72, 0.69, 0.73]

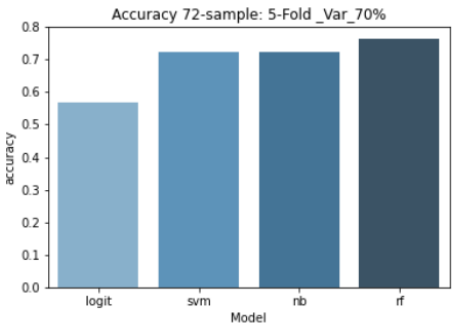


[0.69, 0.73, 0.68, 0.75]

Figure 8: Risultati esperimenti con normalizzazione MinMaxScaler, mantenendo varianza a 80% (16 caratteristiche).

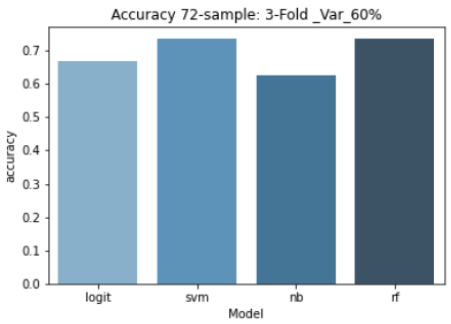


[0.65, 0.76, 0.66, 0.70]

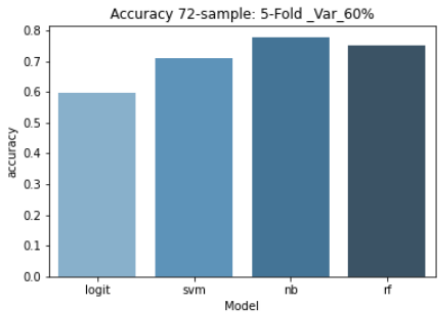


[0.56, 0.72, 0.72, 0.76]

Figure 9: Risultati esperimenti con normalizzazione MinMaxScaler, mantenendo varianza a 70% (11 caratteristiche).

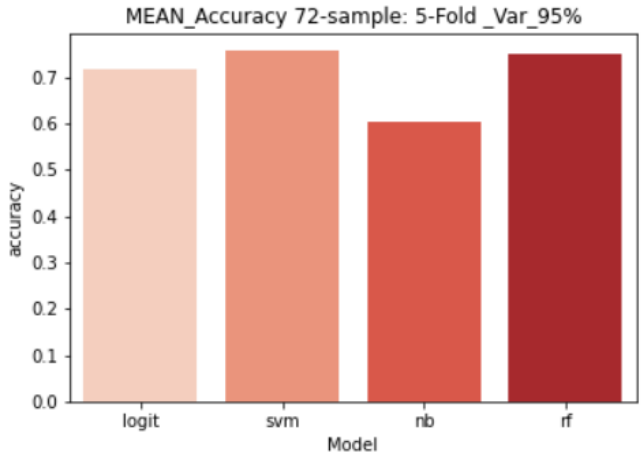


[0.66, 0.73, 0.625, 0.73]



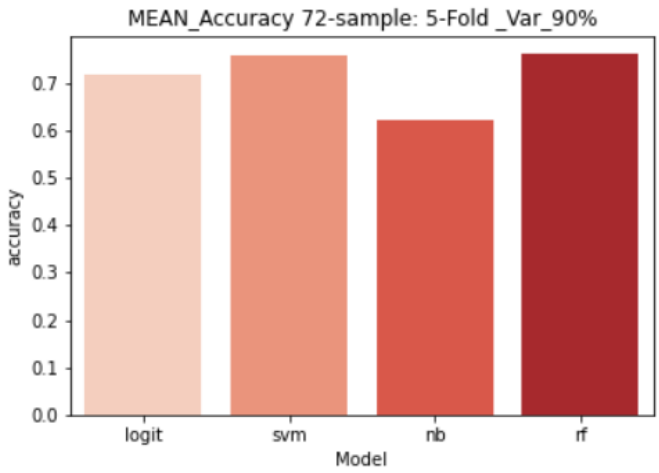
[0.59, 0.70, 0.77, 0.75]

Figure 10: Risultati esperimenti con normalizzazione MinMaxScaler, mantenendo varianza a 60% (9 caratteristiche).



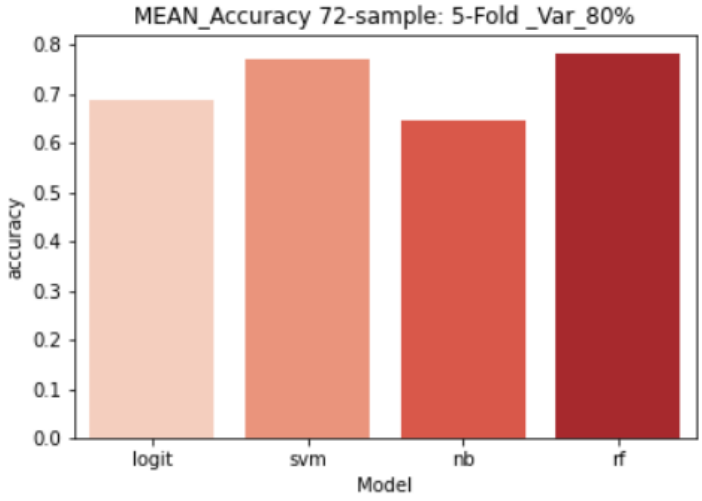
[0.71, 0.75, 0.60, 0.75]

Figure 11: Risultati in media degli esperimenti con PCA (varianza 95%) ripetuti 10 volte.



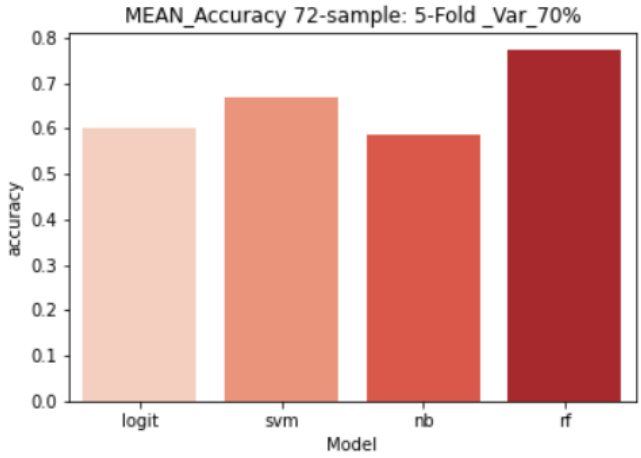

```
[0.71, 0.75, 0.61, 0.76]
```

Figure 12: Risultati in media degli esperimenti con PCA (varianza 90%) ripetuti 10 volte.



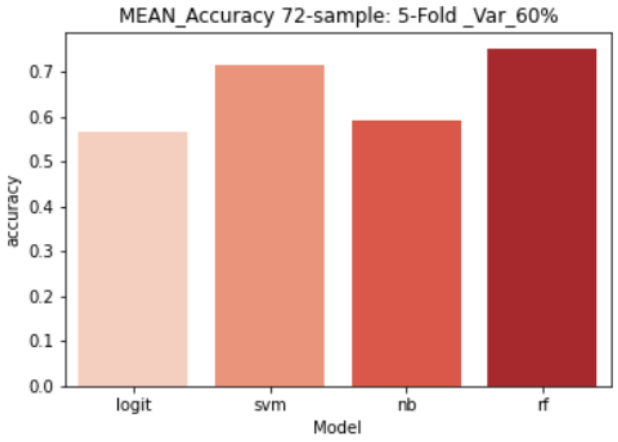

```
[0.68, 0.76, 0.64, 0.78]
```

Figure 13: Risultati in media degli esperimenti con PCA (varianza 80%) ripetuti 10 volte.



[0.6, 0.66, 0.58, 0.77]

Figure 14: Risultati in media degli esperimenti con PCA (varianza 70%) ripetuti 10 volte.



[0.56, 0.71, 0.59, 0.75]

Figure 15: Risultati in media degli esperimenti con PCA (varianza 60%) ripetuti 10 volte.

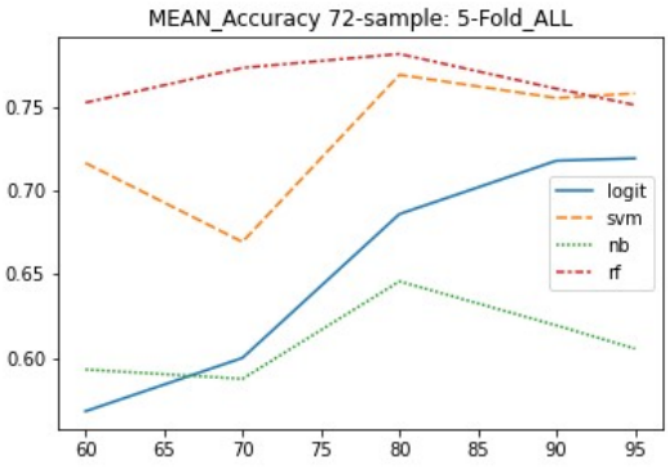


Figure 16: Risultati in media degli esperimenti con PCA.

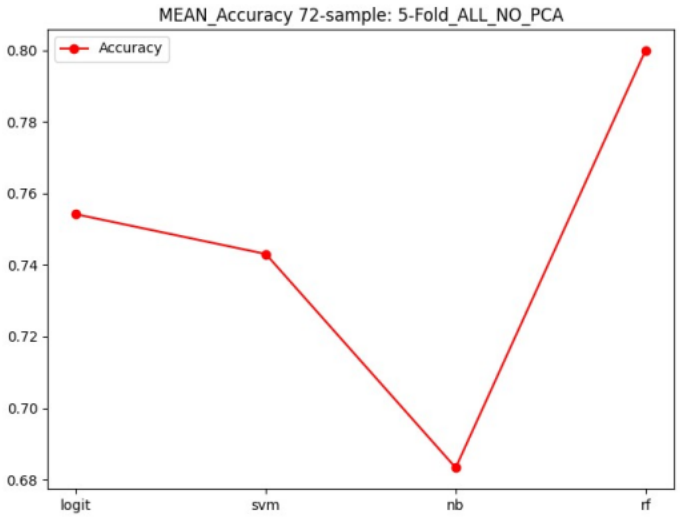


Figure 17: Risultati in media degli esperimenti senza PCA.

A questo punto, si potrebbe pensare che i risultati ottenuti siano stati "falsati" dal fatto che la moltiplicazione tra i samples, per fare il sovraccampionamento, avviene su tutti i dati e quindi in fase di validazione il modello di apprendimento Random Forest predice bene lo Status del tumore poiché in un certo senso i dati utilizzati nell'apprendimento "li ha già visti". Infatti, i dati presenti nella validazione possono essere stati costruiti artificialmente da SMOTE attraverso la moltiplicazione di 2 samples presenti nella fase di apprendimento (training) del modello.

Per questo motivo nella sezione successiva utilizzeremo strategie più "controllate" di sovraccampionamento dei dati (sempre utilizzando la tecnica SMOTE).

3.2. Applicazione strategie di oversampling più accurate

Da questo momento in poi gli esperimenti presentati saranno più accurati, in relazione alla tecnica di sovraccampionamento utilizzata. L'oversampling dei samples è stato effettuato solo sull'insieme utilizzato per l'apprendimento (training) dei modelli. Il set di training sarà composto da un numero di campioni tra 48 e 66 per fold (inizialmente sono 36, i restanti sono dati sintetici, generati con la tecnica SMOTE applicata in base allo Status presente nei samples presi per l' i-esimo fold). Il set di validazione sarà invece composto, sempre, da un numero di campioni pari a 9; Alla base di ogni esperimento viene applicata sempre una normalizzazione dei dati; I test effettuati riportano sempre una valutazione dei risultati sia utilizzando lo StandardScaler (con deviazione standard) (Figure 18, 19) e sia il MinMaxScaler (Figure 20, 21).

Nota: non vengono effettuate, in questo caso, prove con un 3-fold cross-validation perché il generatore di dati sintetici SMOTE non riesce a creare nuovi dati, se il numero di rappresentati per il dato x preso per il k-esimo fold è ≤ 1 (e utilizzando il 3-fold su questo dataset "critico", questo caso si verifica).

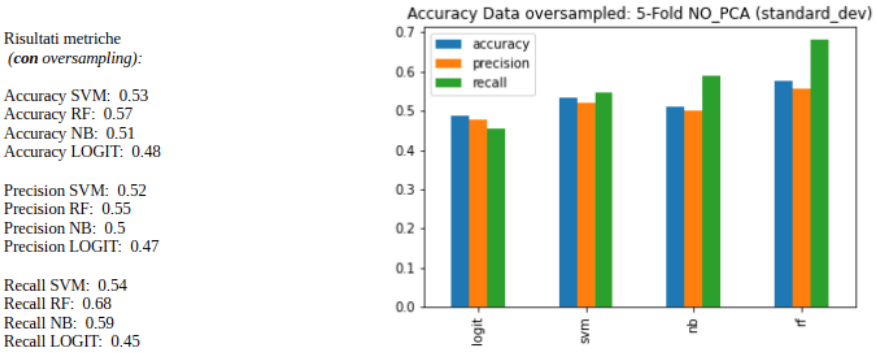


Figure 18: Risultati esperimenti utilizzando set di training sovraccampionato, normalizzazione StandardScaler, senza PCA.

A questo punto, visto che i risultati applicando la tecnica SMOTE non hanno portato significativi miglioramenti rispetto allo stato iniziale degli esperimenti, si è deciso di adottare **2 strategie diverse di oversampling sui dati del traning set**, durante la fase di apprendimento di ogni fold:

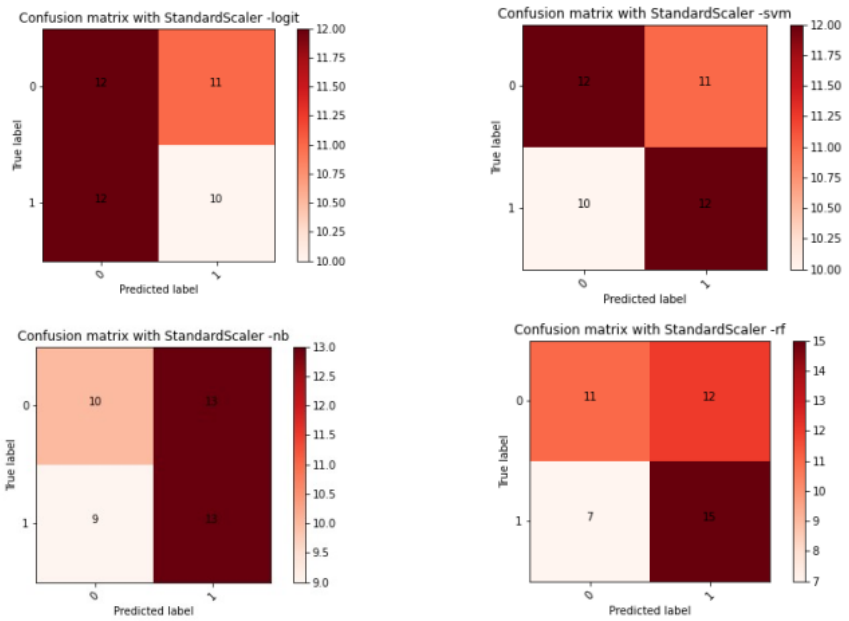


Figure 19: Matrici di confusione: esperimenti utilizzando set di training sovraccampionato, normalizzazione StandardScaler, senza PCA.

Risultati metriche
***(con** oversampling)*:

Accuracy SVM: 0.48
Accuracy RF: 0.64
Accuracy NB: 0.51
Accuracy LOGIT: 0.53

Precision SVM: 0.48
Precision RF: 0.625
Precision NB: 0.5
Precision LOGIT: 0.52

Recall SVM: 0.54
Recall RF: 0.68
Recall NB: 0.59
Recall LOGIT: 0.54

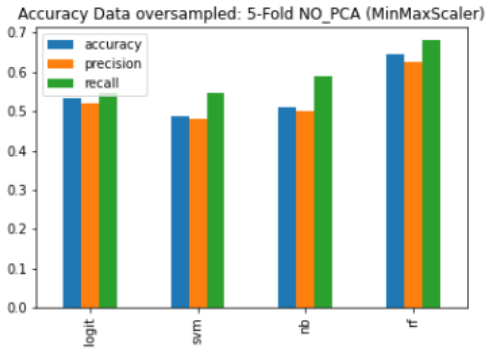


Figure 20: Risultati esperimenti utilizzando set di training sovraccampionato, normalizzazione MinMaxScaler, senza PCA.

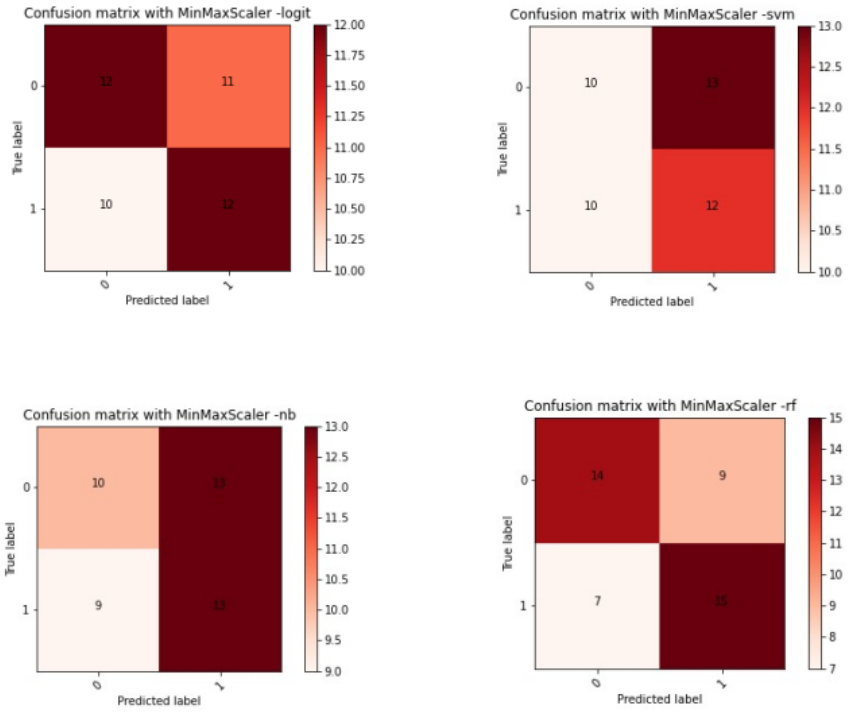


Figure 21: Matrici di confusione: esperimenti utilizzando set di traning sovraccampionato, normalizzazione MinMaxScaler, senza PCA.

1. Nel primo caso (Figure 22, 24, 23, 25), la strategia adottata è quella di fare oversampling del training set in base alla classe più rappresentata tra le 6 micro classi (0,T) (0,N) (0, M) (1,T) (1,N) (1, M);
2. Nel secondo caso (Figure 26, 28, 27, 29), la strategia adottata è quella di fare oversampling bilanciando le micro classi a 2 a 2. Nel fase di apprendimento, nel training set bilanciamo tra di loro le classi:
   - (0,T) con (1,T);
   - (0,N) con (1,N);
   - (0,M) con (1,M);

Nel primo e nel secondo caso, esperimenti sono stati effettuati su 5-fold per ogni run (per un totale di 10 run), facendo oversampling dei campioni (solo sull' insieme di Training nella fase di apprendimento dei modelli) utilizzando la tecnica SMOTE (Synthetic Minority Oversampling Technique).
Nel primo caso: il set di traning sarà composto da un numero di campioni tra 48 e 66 per fold (inizialmente sono 36, i restanti sono dati sintetici generati dall'oversampling). Il set di validazione sarà invece composto, sempre, da un numero di campioni pari a 9;
Nel secondo caso: il set di traning sarà composto da un numero di campioni tra 44 e 50 per fold. Il set di validazione sarà invece composto, sempre, da un numero di campioni pari a 9;
Nel primo e nel secondo caso, compariamo, dunque, i risultati ottenuti sia con l'uso del MinMaxScaler, sia con l'uso dello StandardScaler con deviazione standard. Inoltre differenziamo gli esperimenti in base al "non utilizzo" della PCA e, all'uso della PCA (varianza a 95%).

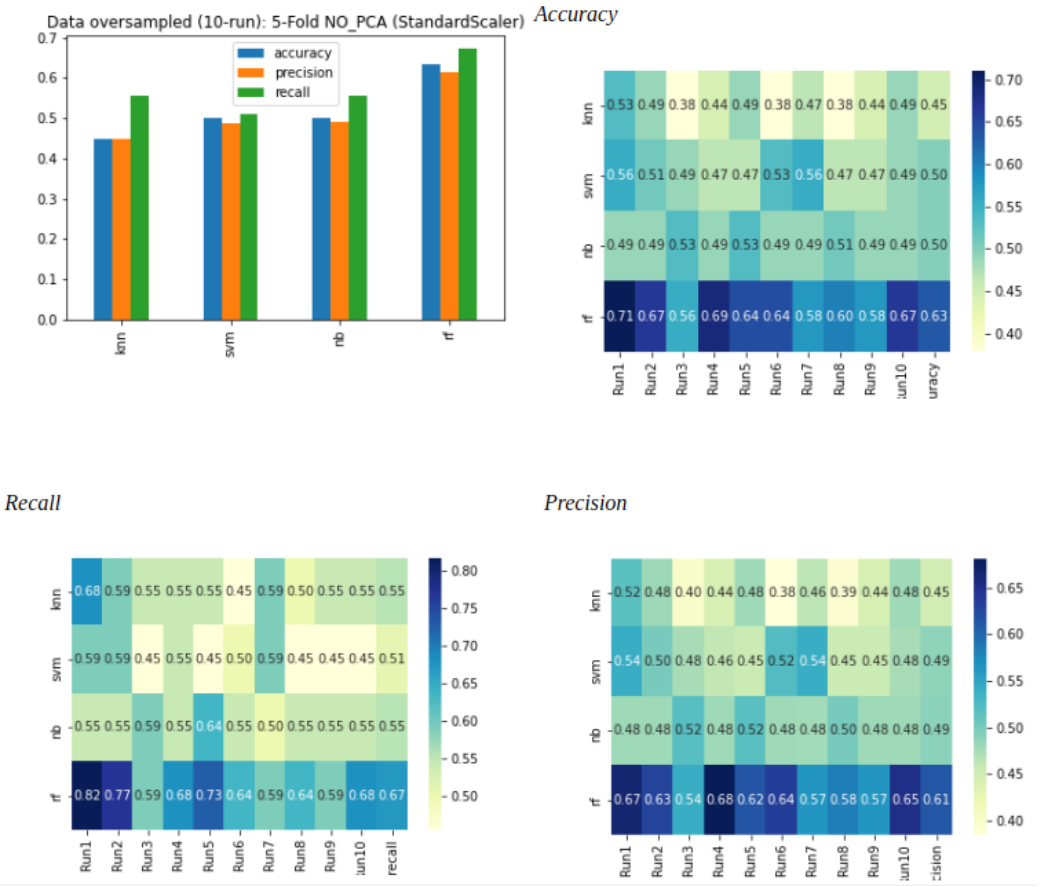


Figure 22: Risultati esperimenti utilizzando la prima strategia di oversampling sul Training set (senza uso della PCA), uso per la standardizzazione dei dati StandardScaler con deviazione standard.
Nota: nell'ultima colonna delle tabelle è presente la media dell'accuracy, della precision e della recall, calcolata su 10 run.

Notiamo che i risultati *"migliori"* si ottengono con l'utilizzo della PCA, in particolare, sia nella prima che nella seconda strategia di oversampling utilizzata. Notiamo che risultati apprezzabili sono, nello specifico, osservabili con l'utilizzo della normalizzazione dei dati con MinMaxScaler (sia con la prima, che con la seconda strategia).

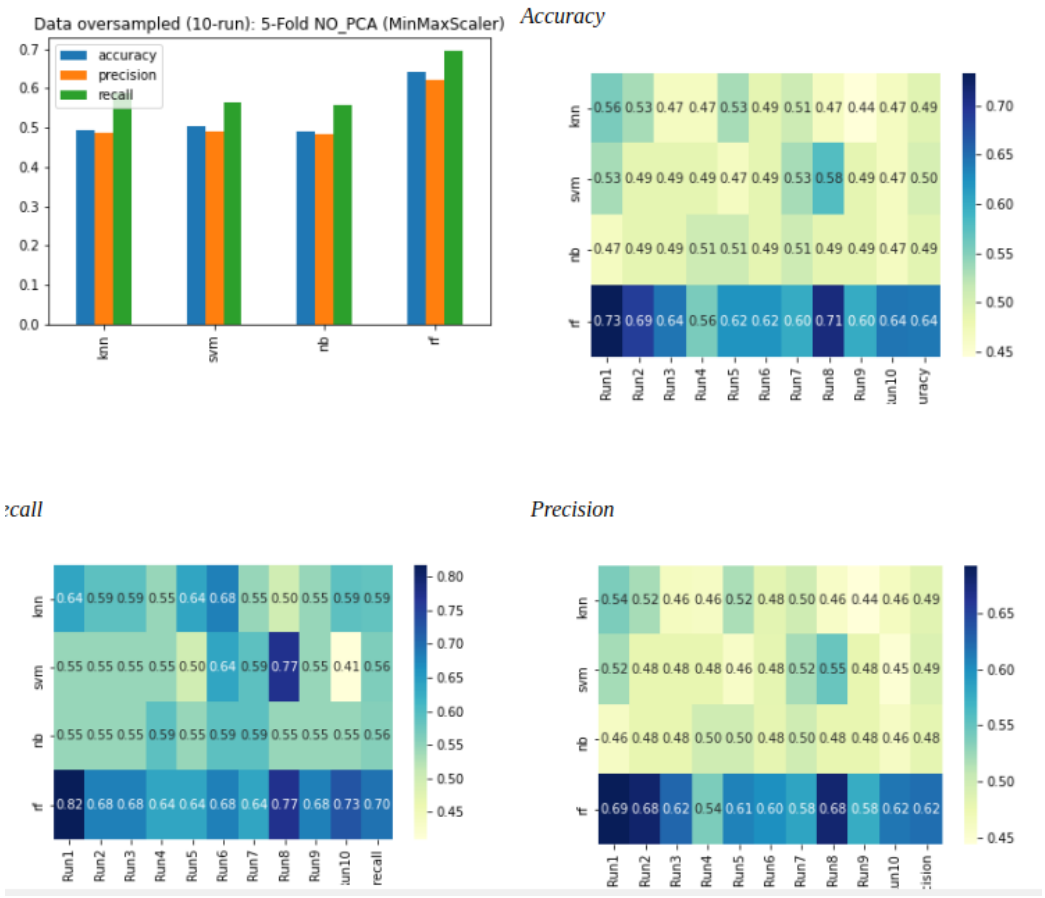


Figure 23: Risultati esperimenti utilizzando la prima strategia di oversampling sul Training set (senza uso della PCA), uso per la standardizzazione dei dati MinMaxScaler.
Nota: nell'ultima colonna delle tabelle è presente la media dell'accuracy, della precision e della recall, calcolata su 10 run.

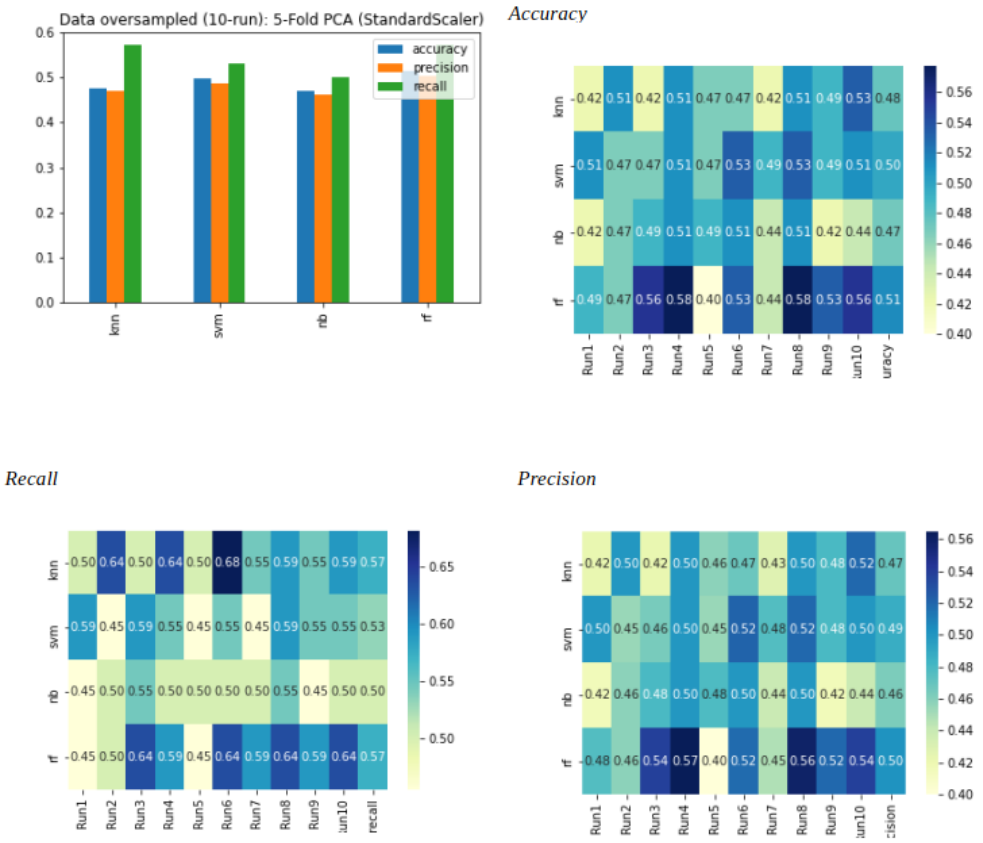


Figure 24: Risultati esperimenti utilizzando la prima strategia di oversampling sul Training set (con uso della PCA), uso per la standardizzazione dei dati StandardScaler con deviazione standard.
Nota: nell'ultima colonna delle tabelle è presente la media dell'accuracy, della precision e della recall, calcolata su 10 run.

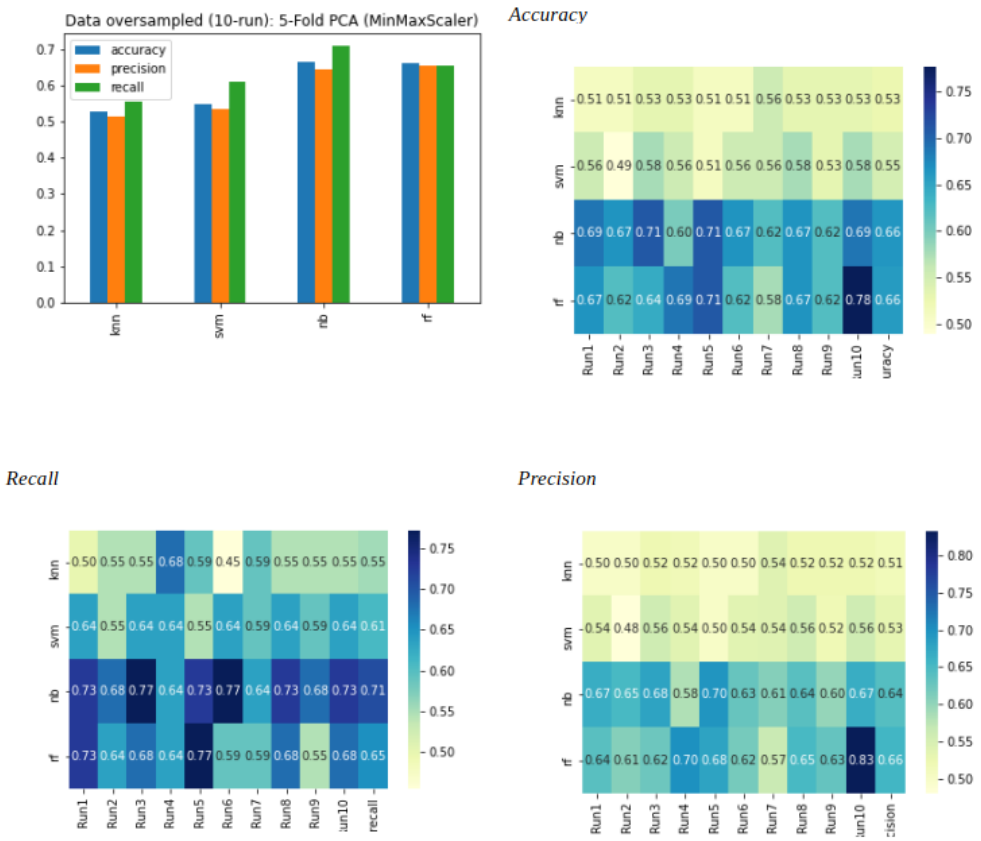


Figure 25: Risultati esperimenti utilizzando la prima strategia di oversampling sul Training set (con uso della PCA), uso per la standardizzazione dei dati MinMaxScaler.
Nota: nell'ultima colonna delle tabelle è presente la media dell'accuracy, della precision e della recall, calcolata su 10 run.

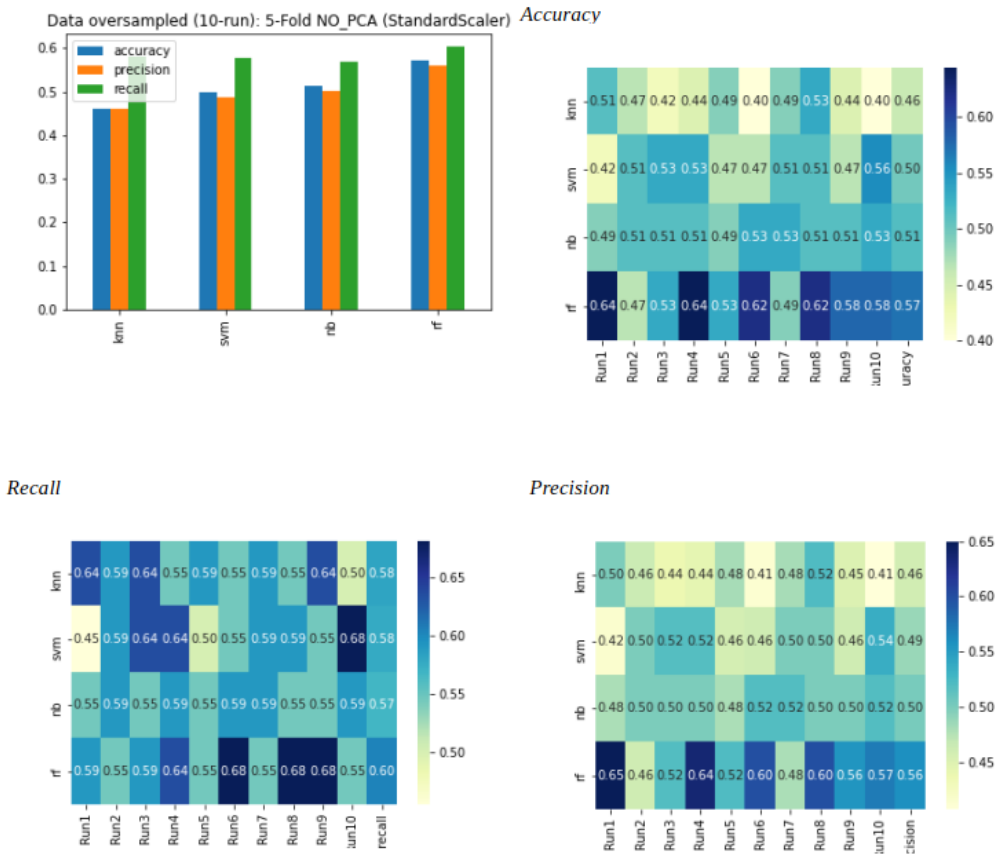


Figure 26: Risultati esperimenti utilizzando la seconda strategia di oversampling sul Training set (senza uso della PCA), uso per la standardizzazione dei dati StandardScaler con deviazione standard.
Nota: nell'ultima colonna delle tabelle è presente la media dell'accuracy, della precision e della recall, calcolata su 10 run.

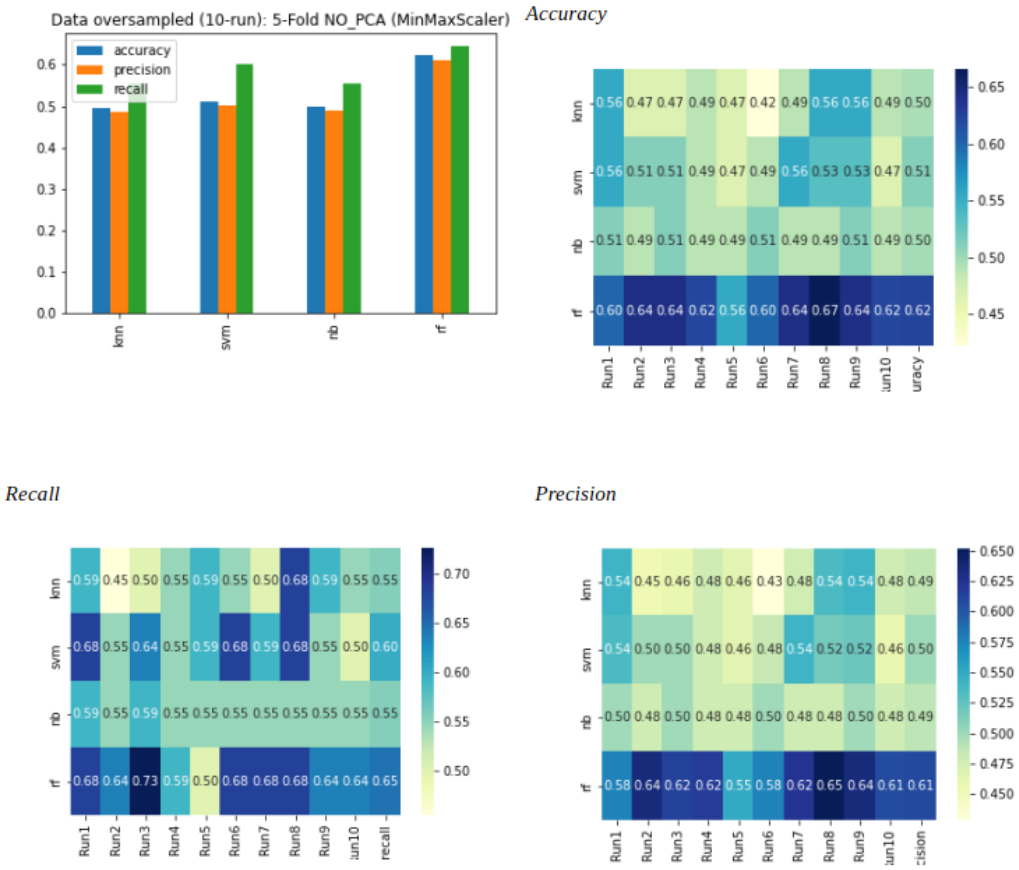


Figure 27: Risultati esperimenti utilizzando la seconda strategia di oversampling sul Training set (senza uso della PCA), uso per la standardizzazione dei dati MinMaxScaler.
Nota: nell'ultima colonna delle tabelle è presente la media dell'accuracy, della precision e della recall, calcolata su 10 run.

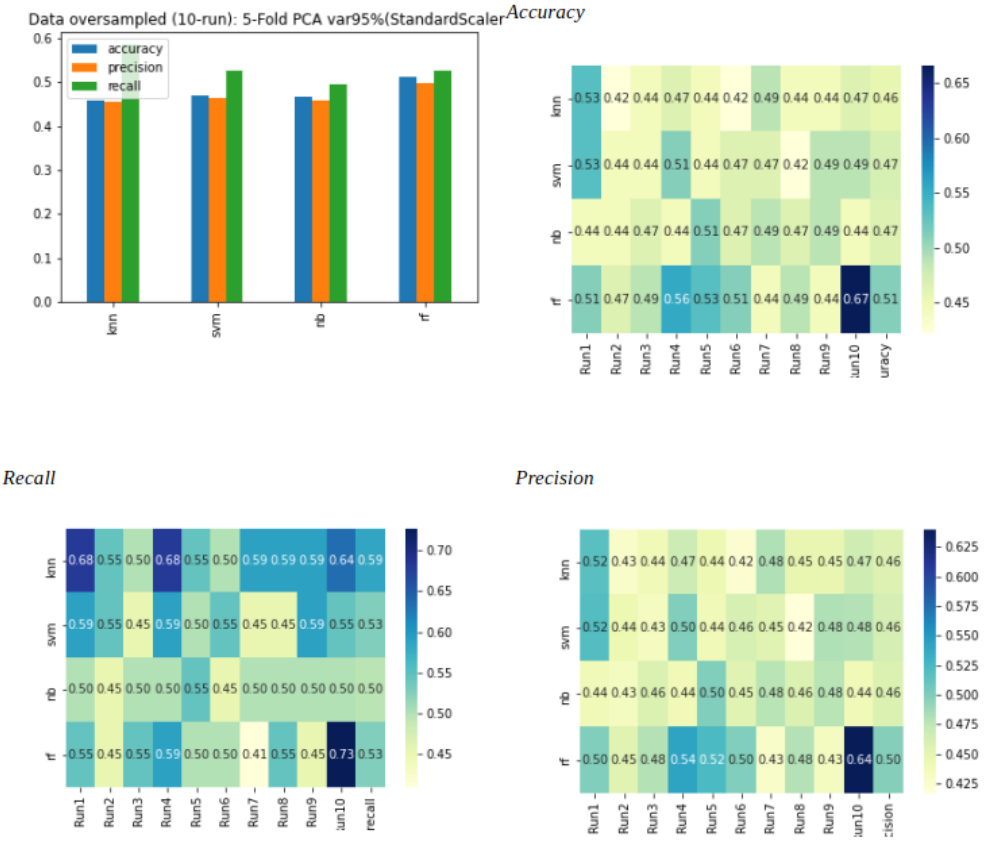


Figure 28: Risultati esperimenti utilizzando la seconda strategia di oversampling sul Training set (con uso della PCA), uso per la standardizzazione dei dati StandardScaler con deviazione standard.
Nota: nell'ultima colonna delle tabelle è presente la media dell'accuracy, della precision e della recall, calcolata su 10 run.

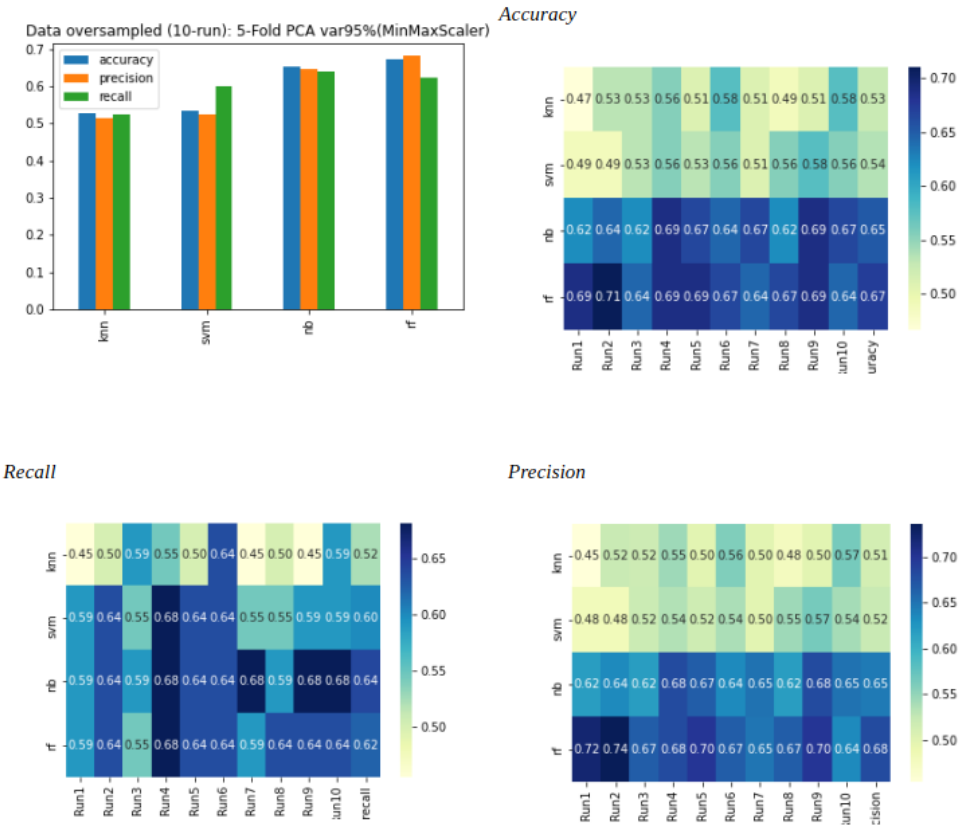


Figure 29: Risultati esperimenti utilizzando la seconda strategia di oversampling sul Training set (con uso della PCA), uso per la standardizzazione dei dati MinMaxScaler.
Nota: nell'ultima colonna delle tabelle è presente la media dell'accuracy, della precision e della recall, calcolata su 10 run.

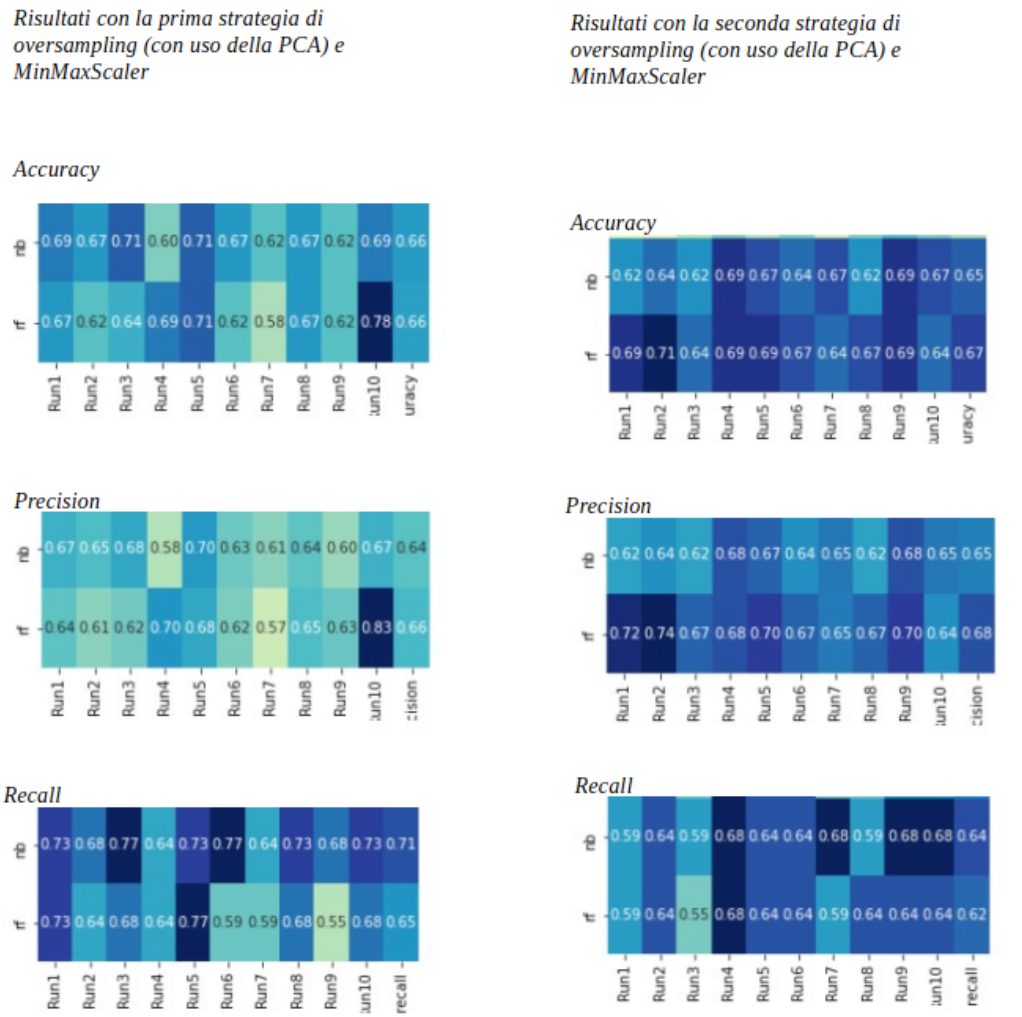


Figure 30: Presentiamo una comparazione, rappresentando solo i risultati del modello Naive Bayes e Random Forest, i quali generano risultati migliori rispetto agli altri modelli utilizzati.
NOTA: I risultati a destra presentano un colore più uniforme, perché si riescono ad ottenere risultati, per qualsiasi esecuzione random del programma, più o meno simili tra di loro per ogni run effettuato (dunque predizioni meno randomiche).

4. Introduzione agli autoencoder

Gli autoencoder sono un tipo specifico di reti neurali feedforward di tipo deep, in cui l'ingresso è uguale all'uscita. Un autoencoder è costituito da 3 componenti: encoder, codifica dei dati e decoder. Gli autoencoders hanno il compito di comprimere l'ingresso in uno spazio di rappresentazione a di dimensioni inferiori e poi di ricostruire l'uscita a partire da questa rappresentazione "ridotta". Quest'ultima è il risultato della *compressione* dell'input, chiamata anche rappresentazione dello spazio latente, generata dall'encoder. Lo scopo finale, è trovare il modello d'autoencoder che mantenga il massimo di informazioni durante la codifica e, quindi, abbia il minimo errore di ricostruzione durante la decodifica .

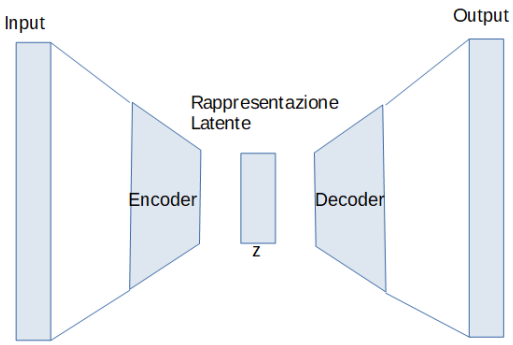


Figure 31: Struttura Autoencoder

Negli esperimenti utilizzando un particolare autoencoders, in un certo senso andiamo a sostituire ciò che precedentemente facevamo con la PCA. La riduzione della dimensionalità può essere interpretata come compressione dei dati in cui il codificatore comprime i dati (dallo spazio iniziale allo spazio spazio latente) mentre il decodificatore li decomprime.

In tale situazione, possiamo vedere un chiaro collegamento con la PCA nel senso che, proprio come fa la PCA, stiamo cercando il miglior sottospazio lineare su cui proiettare i dati con la minor perdita di informazioni possibile quando lo facciamo. La ricerca di encoder e decoder che minimizzino l'errore di ricostruzione avviene per discesa del gradiente sui parametri di queste reti.

In effetti, è possibile scegliere diverse basi per descrivere lo stesso sottospazio ottimale, quindi, diverse coppie codificatore / decodificatore possono fornire l'errore di ricostruzione ottimale. Inoltre, per gli autoencoder lineari e contrariamente alla PCA, le nuove funzionalità non devono essere

indipendenti (nessun vincolo di ortogonalità nelle reti neurali). La mancanza di struttura tra i dati codificati nello spazio latente è abbastanza normale. L'autoencoder è addestrato unicamente a codificare e decodificare con il minor numero di perdite possibile, indipendentemente da come sia organizzato lo spazio latente. Il problema dell'utilizzare un autoencoder "semplice" sta proprio nel modo di rappresentare la compressione dei dati in input (senza struttura), e ciò può provocare la possibilità di overfitting, a meno che non si faccia uso di una particolare regolarizzazione.

5. Utilizzo del modello VAE e vantaggi

La regolarità dello spazio latente per gli autoencoder è un punto difficile che dipende dalla distribuzione dei dati nello spazio iniziale, dalla dimensione dello spazio latente e dall'architettura dell'encoder. Quindi, è piuttosto difficile (se non impossibile) garantire, a priori, che l'encoder organizzi lo spazio latente in modo intelligente compatibile con il processo generativo.

I modelli generativi profondi hanno mostrato un'incredibile capacità di produrre contenuti altamente realistici di vario tipo, come immagini, testi e suoni. Tra questi modelli generativi profondi, due grandi famiglie si distinguono e meritano un'attenzione speciale: Generative Adversarial Networks (GANs) e Variational Autoencoders (VAEs).

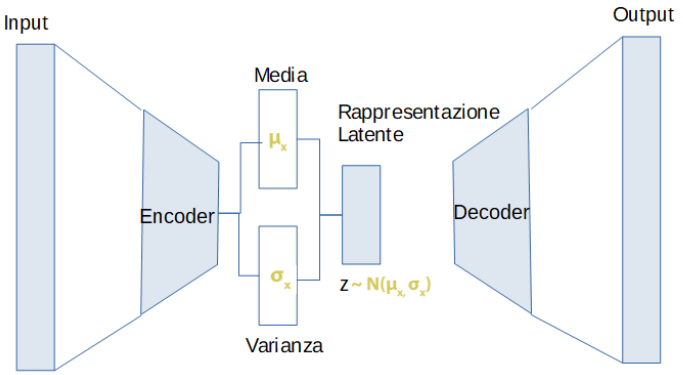


Figure 32: Struttura Variational Autoencoder (VAE)

I Variational Autoencoders (VAE) codificano gli input come distribuzioni invece di semplici punti. Un VAE è un particolare tipo di autoencoder, la cui distribuzione delle codifiche viene regolarizzata durante l'addestramento in

modo da garantire che il suo spazio latente abbia proprietà tali da permettere di generare nuovi dati a partire da esso (processo generativo).

La regolarità dello spazio latente per rendere possibile il processo generativo deve garantire che la decodifica di due punti vicini nello spazio non deve avere risultati completamente discordanti, e ogni punto campionato nello spazio latente deve avere un "significato" in output. La regolarizzazione viene eseguita imponendo che le distribuzioni siano vicine a una distribuzione normale standard, richiedendo che le matrici di covarianza siano vicine all'identità, e impedendo che le distribuzioni codificate siano troppo distanti l'una dall'altra.

Figure 33: A sinistra un esempio di cosa accade senza la regolarizzazione. A destra, un esempio di struttura regolarizzata.

Al fine di utilizzare i variational autoencoders per estrarre le caratteristiche rilevanti dai samples e riuscire a classificare, in base allo "Status" del tumore, correttamente qualsiasi input, utilizziamo l'architettura in Figure 34.

L'esperimento è stato cosi condotto:

- A partire dal dataset iniziale (45 sample) si utilizza un Stratifield 9-fold, per la suddivisione del dataset di "Traning" e del dataset di Test, ad ogni nuovo fold con l'estrazione randomica dei campioni.
- A questo punto, il dataset di "Training" precedentemente generato in maniera casuale viene passato ad un Stratifield 5- fold, per la suddivisione dei dati in: set di Traning, e set di Validation. Utilizzati per l'apprendimento del Variational Autoencoder.

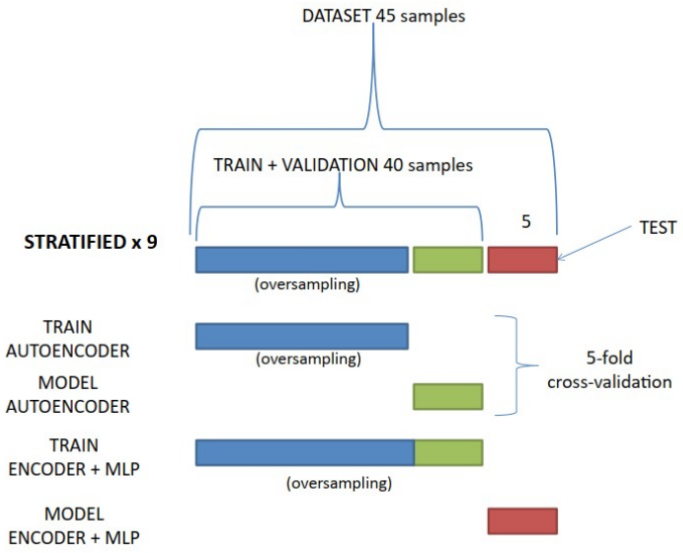


Figure 34: Struttura del modello di classificazione utilizzando VAE

- Infine, riutilizzando il dataset "Training+Validation", effettuiamo l'apprendimento della rete fully connected creata, collegata all'encoder del VAE.
  Nota: in questa fase i layers utilizzati dall'encoder devono essere prima "spenti" (settati a False). Poi in un secondo momento, devono essere "riaccesi" (settati a True), per effettuare l'apprendimento sulla rete Encoder + MPL.

- Nella fase finale, valutiamo le predizioni della rete e dunque le metriche (Accuracy, Precision, Recall) solo sul dataset di Test, creato inizialmente.

  Sono stati effettuati esperimenti anche facendo oversampling solo sul set di Traning all'interno della fase 5-fold cross validation, utilizzando SMOTE e facendo oversampling in riferimento alla caratteristica "Status" dei sample (Figure 35, 36). Ulteriori esperimenti, sono stati effettuati provando ad utilizzare la rete VAE come modello generativo di dati sintetici (Figure 37, 38, 39).

6. Risultati

Di seguito vengono riportati tutti gli esperimenti effettuati, utilizzando il modello Variational Autoencoder (VAE).

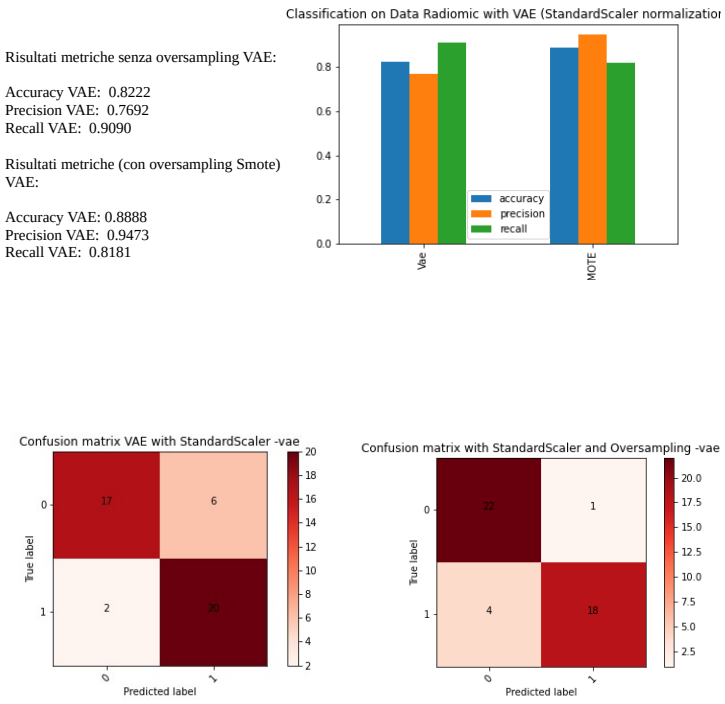


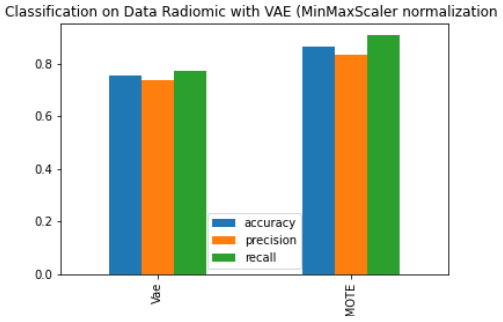
Figure 35: Risultati esperimenti con classificatore VAE+MPL: normalizzazione StandardScaler, senza/con oversampling (SMOTE) sul training set.

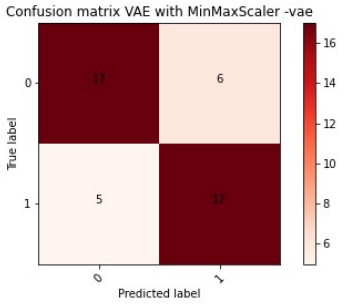
Risultati metriche senza oversampling VAE:

Accuracy VAE: 0.7555
Precision VAE: 0.7391
Recall VAE: 0.7727

Risultati metriche (con oversampling Smote) VAE:

Accuracy VAE: 0.8666
Precision VAE: 0.8333
Recall VAE: 0.9090





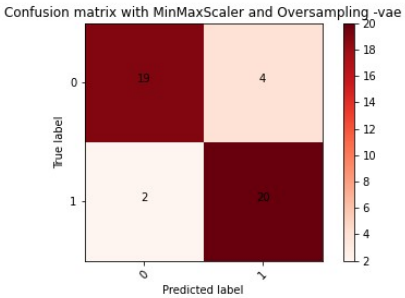


Figure 36: Risultati esperimenti con classificatore VAE+MPL: normalizzazione MinMaxScaler, senza/con oversampling (SMOTE) sul training set.

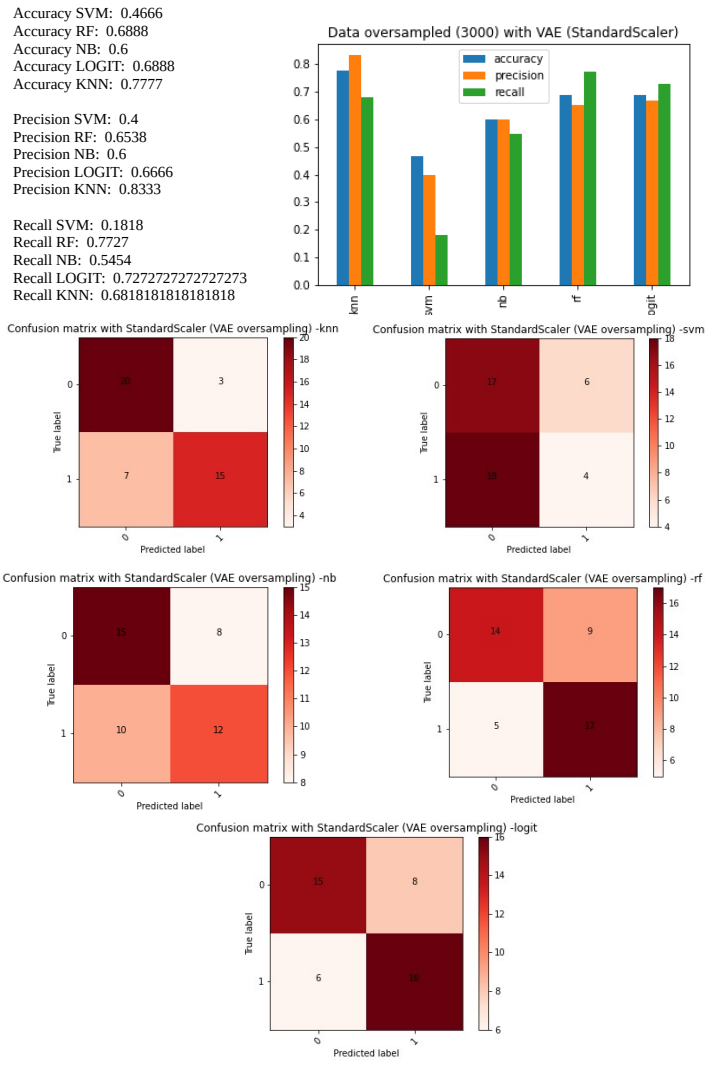


Figure 37: Risultati esperimenti usando come classificatore i modelli ML (RF, KNN, NB, SVM, LOGIT): normalizzazione StandardScaler, con oversampling usando VAE come modello generativo di dati sintetici sul training set.

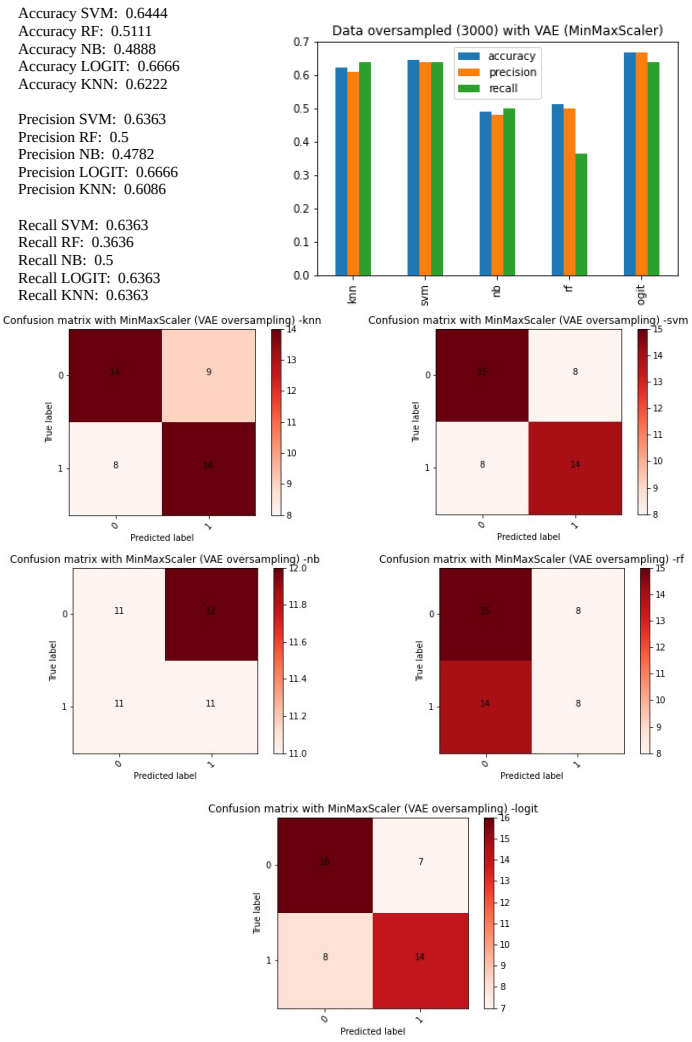


Figure 38: Risultati esperimenti usando come classificatore i modelli ML (RF, KNN, NB, SVM, LOGIT): normalizzazione MinMaxScaler, con oversampling usando VAE come modello generativo di dati sintetici sul training set.

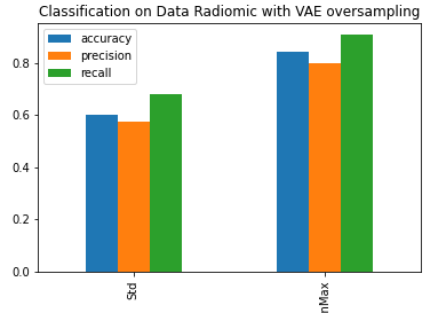
Risultati metriche:

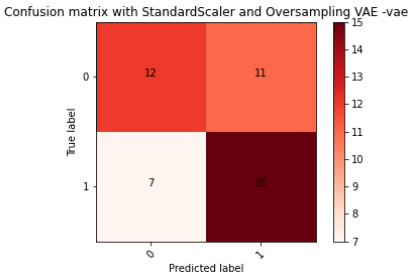
Con StandardScaler:

Accuracy VAE: 0.6
Precision VAE: 0.57
Recall VAE: 0.68

MinMaxScaler:

Accuracy VAE: 0.84
Precision VAE: 0.8
Recall VAE: 0.90





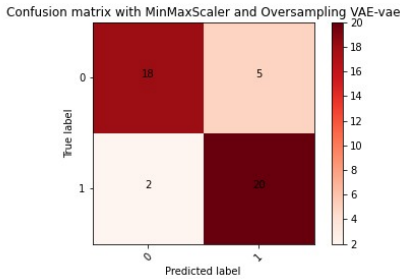


Figure 39: Risultati esperimenti usando come classificatore la rete MPL: normalizzazione MinMaxScaler e StandardScaler, con oversampling usando VAE come modello generativo di dati sintetici sul training set.

7. Conclusioni

A seguito dei risultati ottenuti negli esperimenti finali, possiamo affermare che un certo miglioramento apprezzabile è osservabile grazie all'utilizzo dei Variational Autoencoders. In particolare si nota, che il modello VAE riesce a selezionare meglio le caratteristiche dei samples, dunque s'addestra meglio sui dati in input rispetto a quanto non fanno i modelli "semplici" di ML con l'uso della PCA. Difatti, comparando i risultati ottenuti utilizzando VAE come modello generativo di dati sintetetici, si nota un miglioramento delle predizioni rispetto all'uso di PCA e SMOTE, soprattutto nel caso di :

- KNN, RF, LOGIT con utilizzo della standardizzazione Standard Scaler;
- KNN, SVM, LOGIT con utilizzo della standardizzazione MinMax Scaler;

In generale, dagli esperimenti eseguiti utilizzando VAE al fine di apprendere e classificare, i risultati ottenuti sono molto incoraggianti. Infatti, per quanto riguarda l'uso della di standardizzazione dei dati StandardScaler (Figure 35) otteniamo:

- risultati superiori all'80% per le metriche Accuracy, Precision, e Recall.
- nel caso dell'uso di oversampling, si riesce ad ottenere un accuracy pari a 88% e una precision del 94%.

Per quanto riguarda, invece, l'uso della di standardizzazione dei dati MinMaxScaler (Figure 36) otteniamo:

- risultati intorno all'75% per le metriche Accuracy, Precision, e Recall.
- nel caso dell'uso di oversampling, si riesce ad ottenere un accuracy pari a 86% e una precision del 83%.

Durante gli esperimenti con i Variational Autoencoders si è provato a fare l'oversampling dei campioni, prendendo come riferimento la caratteristica "Tipologia" (T, N, M) del tumore, in modo da effettuare diverse strategie di oversampling sul set di training, cosi come è stato fatto precedentemente. In questo caso, siccome per ogni fold i samples erano in numero inferiore (suddividiamo il dataset iniziale in traning, validation e test), gli esperimenti non sono andati a "buon fine" poiché non si è riuscito a moltiplicare e dunque creare dati sintetici per una particolare tipologia del tumore (M), in cui il

numero di rappresentati presi per il k-esimo fold è ≤ 1. In particolare, il problema è sui samples con Status 0, e tipologia M (solo 3 samples rappresentati in tutto il dataset). Probabilmente se non consideriamo i samples (sia con status 0 e 1) per la tipologia di tumore M, le predizioni darebbero risultati ancora più alti. Questo, significherebbe però ridurre ulteriormente il dataset già fortemente esiguo, a un numero di campioni pari a 33. L'ideale sarebbe applicare il VAE a un dataset più ampio, in modo da effettuare esperimenti ancor più approfonditi, rispetto a quanto è stato già fatto.

References

[1] Coroller TP, Grossmann P, Hou Y, Rios Velazquez E, Leijenaar RT, Hermann G, et al. (March 2015). "CT-based radiomic signature predicts distant metastasis in lung adenocarcinoma". Radiotherapy and Oncology. 114 (3): 345–50. doi:10.1016/j.radonc.2015.02.015. PMC 4400248. PMID 25746350.

[2] Pierpaolo Alongi, Riccardo Laudicella, Alessandro Stefano, Federico Caobelli, Albert Comelli, Antonio Vento, Davide Sardina, Gloria Ganduscio, Patrizia Toia, Francesco Ceci, Paola Mapelli, Maria Picchio, Massimo Midiri, Sergio Baldari, Roberto Lagalla, Giorgio Russo. "Choline PET/CT features to predict survival outcome in high risk prostate cancer restaging: a preliminary machine-learning radiomics study".

[3] https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73