Elaborato di Machine Learning

Anderloni Hanan Francesca (mat. 889079)

Finzi Micol Rebecca (mat. 882523)

Melchioretto Nicolò (mat. 893145)

Pellicani Chiara (mat. 889904)

Data set

https://www.kaggle.com/datasets/radheshyamkollipara/bank-customer-churn/data

Il Dataset che è stato usato contiene informazioni riguardanti i dati dei clienti di una banca. L'obiettivo è prevedere l'abbandono della banca da parte dei clienti.

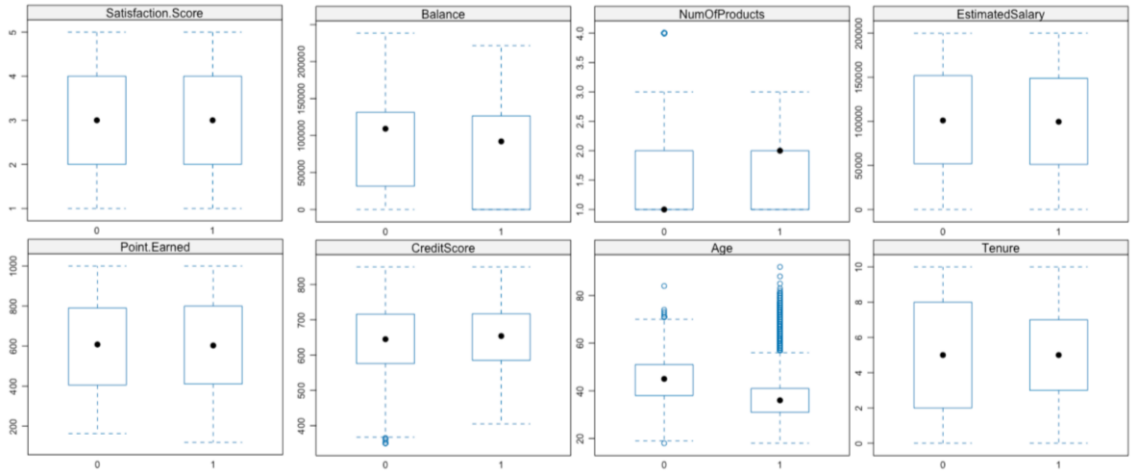
Le variabili del dataset sono:

RowNumber corrisponde al numero del record riga
CustomerId contiene il codice ID del cliente
Surname indica il cognome del cliente
CreditScore indica il punteggio di credito del cliente
Geography indica il paese del cliente, tra Germania, Spagna e Francia
Gender indica il genere del cliente
Age indica l'età della banca
Tenure si riferisce al numero di anni in cui il cliente è stato cliente della banca
Balance indica il saldo nel conto del cliente
NumOfProducts si riferisce al numero di prodotti che un cliente ha acquistato tramite la banca
HasCrCard indica se un cliente ha o meno una carta di credito
IsActiveMember indica se il cliente è attivo o meno
EstimatedSalary indica il salario stimato del cliente
Exited indica se il cliente ha lasciato o meno la banca
Complain indica se il cliente ha un reclamo o meno
SatisfactionScore indica il punteggio fornito dal cliente per la risoluzione del reclamo
CardType indica il tipo di carta trattenuta dal cliente
PointsEarned indica i punti guadagnati dal cliente per l'utilizzo della carta di credito

E' stata usata come target la variabile *Exited*, dove 0 indica se un cliente lascia la banca(evento) e 1 se invece sceglie di restare(non evento).

Gestione del dataset

Sono state subito rimosse le variabili relative all' ID: *RowNumber*, *CustomerID* e *Surname*. Successivamente è stata osservata la distribuzione delle variabili rispetto a quella dipendente, *Exited*.



Il dataset non è stato bilanciato in quanto presenta una distribuzione degli eventi pari al 20%, corrispondente ai valori della popolazione. Inoltre gli eventi non sono rari, quindi non è necessario trattarli (fonte 1, fonte 2, fonte 3).

0	1
20.39035	79.60965

Dopo aver controllato che non ci fossero valori mancanti, è stata controllata la collinearità ed è stata eliminata la variabile *Complain,* in quanto variabile dipendente endogena.

19 19	Exited	Complain	9907.907	1	0.000	10000	1	1	10000	0.99079070359

	VIF	TOL
X_matrixCreditScore	1.0004	0.9996
X_matrixAge	1.0015	0.9985
X_matrixTenure	1.0006	0.9994
X_matrixBalance	1.1031	0.9065
X_matrixNumOfProducts	1.1036	0.9061
X_matrixEstimatedSalary	1.0008	0.9992
X_matrixSatisfaction.Score	1.0008	0.9992
X_matrixPoint.Earned	1.0007	0.9993

Successivamente è stato verificato se fossero presenti fenomeni di *near-zero variance* e *quasi-separation*, ma è risultato non esserci alcun problema sotto questi aspetti.

Dopo aver estratto i dati di score (10% dell'originale), il dataset è stato suddiviso in dati di *training* e di *validation*.
Si è scelto di allenare i modelli che non necessitano di preprocessing - *Random Forest, Tree, Gradient Boosting* e *Bagging* - sui dati di training intoccati, e i rimanenti su dati su cui è stata effettuata model selection tramite l'algoritmo Boruta, con ritocchi *ad hoc* dove necessario sfruttando il comando *preProcess* integrato in *caret*.

Scelta della metrica di tuning

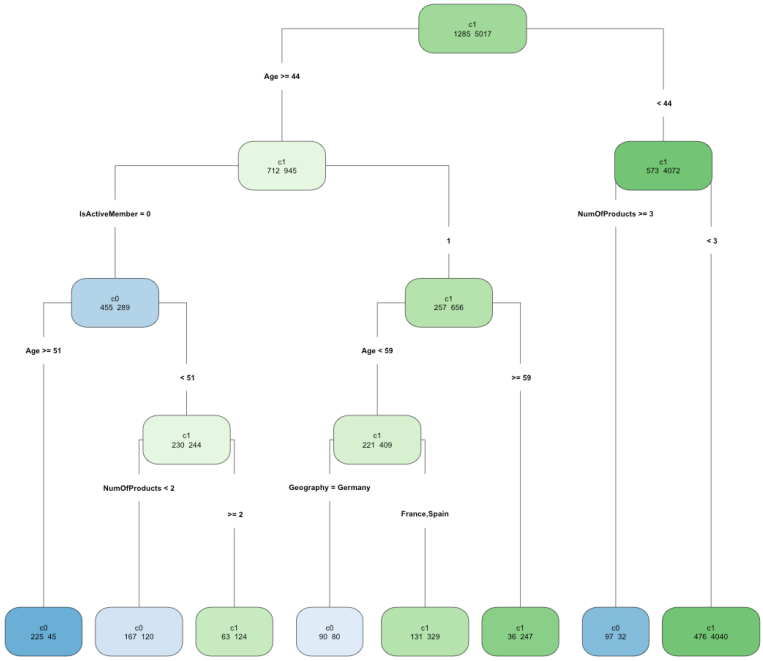
Si è deciso di tunare i modelli sulla Sensitivity $TP/(TP * FN)$: il task classificativo avrà infatti l'obiettivo di identificare correttamente i clienti più propensi a interrompere il contratto con la banca (*TP*) così da consentire a quest'ultima di agire di conseguenza con offerte o altro. In particolare, minimizzare i falsi negativi, vale a dire i soggetti che il modello classifica come "non uscenti" ma che invece sciolgono il contratto.

Il processo di tuning è stato effettuato usando *Repeated Cross-Validation*, che è stata preferita alla *10-fold CV* per il suo bias, inferiore rispetto a quest'ultima, a parità di variabilità.

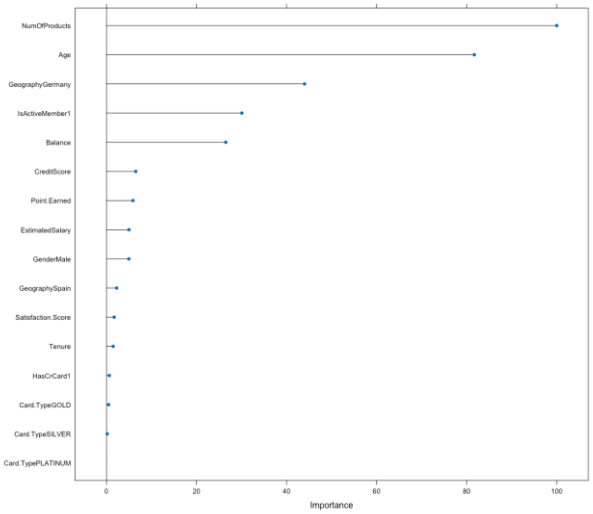
Step 1 - Allenamento dei modelli

Modelli che non richiedono preprocessing

Tree

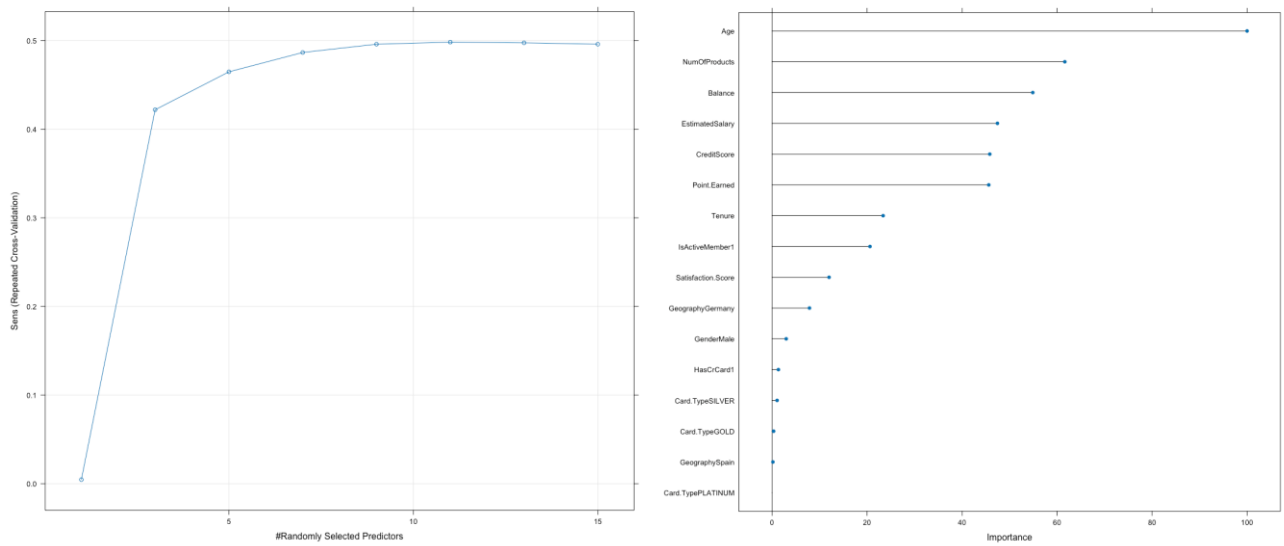


L'albero in questione è stato prunato, in quanto le performance sono pressoché identiche rispetto a quello *full-depth* e indubbiamente ne giova la sua visualizzazione.
Notiamo che gli split più importanti sono generati dalle variabili relative a età del cliente, numero di prodotti detenuti, livello di attività e area geografica di provenienza; rilevanza confermata dalla *variable importance* fornita dall'albero e che persisterà anche negli altri modelli.



Random Forest

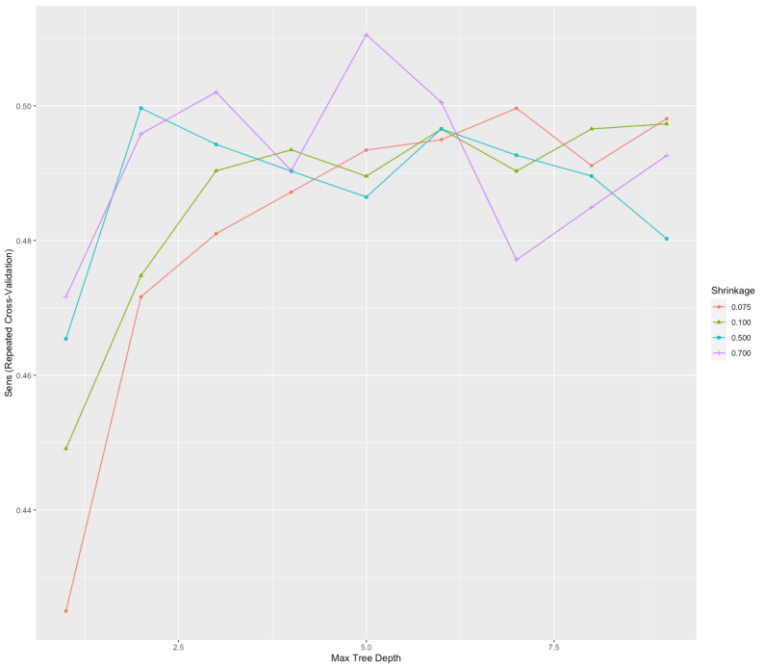
Il training della Random Forest ha prodotto come parametro di tuning crossvalidato un'ampiezza dei subset di variabili per ogni campione bootstrap pari a 11. Di seguito il grafico relativo a tuning e all'importanza delle variabili.



Si noti come le variabili importanti
per RF sono differenti rispetto alla valutazione prodotta dall'albero precedente.

Gradient Boosting

Per l'algoritmo di Gradient Boosting Machine viene riportato il grafico relativo ai quattro parametri di tuning, i cui valori crossvalidati sono risultati essere:



n.trees = nr. di iterazioni (un albero per iterazione) = 200 ;

interaction.depth = profondità degli alberi = 5 ;

shrinkage = learning rate, velocità di apprendimento = 0.7

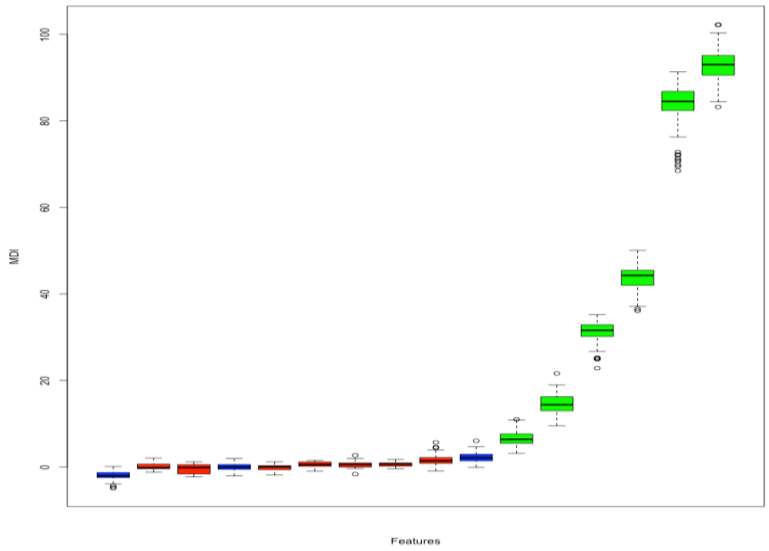
n.minobsinnode = minimo numero di osservazioni necessarie a consentire uno split = 20

Bagging

L'algoritmo Bagging è stato allenato con risultati analoghi ai precedenti modelli.

Preprocessing: model selection

Come si diceva più sopra, per questi modelli è stato utilizzato un dataset su cui è stata eseguita model selection tramite Boruta, che ha sfoltito l'insieme di variabili di partenza. Di seguito il grafico generato dall'algoritmo:

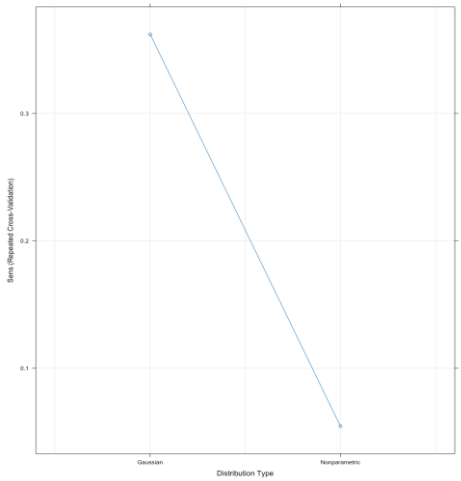


Sono state mantenute 7 variabili, segnate in verde, mentre 6 variabili segnate in rosso sono state invece rimosse, ed è stato creato un nuovo dataset con le covariate selezionate.
Le covariate ottenute con Boruta sono: *Geography, Gender, Age, Balance, NumOfProducts, Is Active Member*.

Tra i modelli che richiedono preprocessing si riportano i risultati dei più rilevanti, poiché PLS, Lasso e logistico si sono rivelati scarsamente performanti.

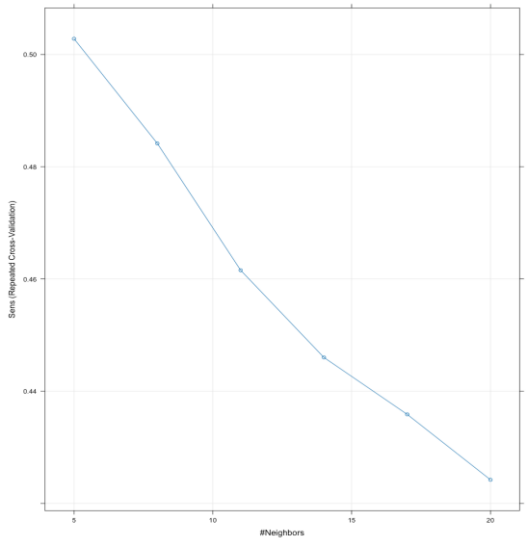
Modelli che richiedono model selection

Naive Bayes



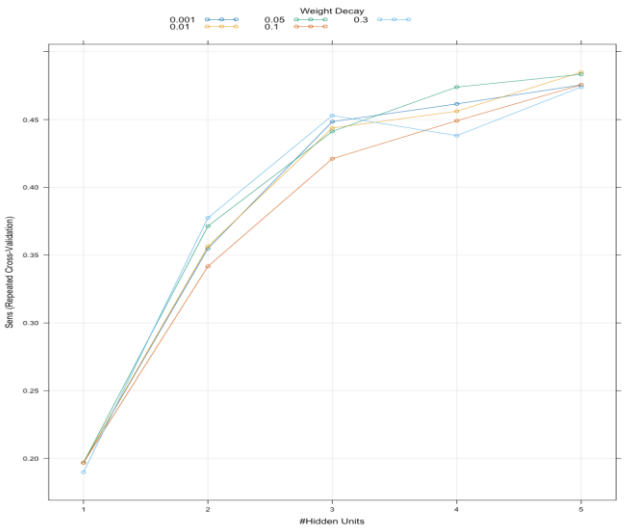
L'algoritmo Naive Bayes ha richiesto la verifica di assenza di missing, 0-problem, near zero variance (già corretti nel dataset iniziale) e della collinearità.
La densità impiegata dall'algoritmo è quella gaussiana.

K-nearest neighbors



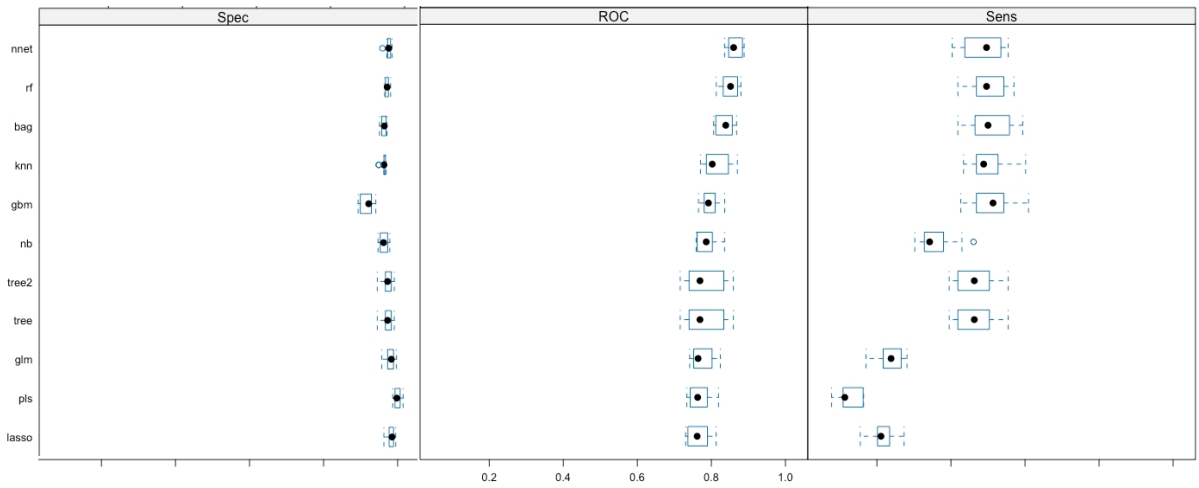
Il k-nearest neighbor richiede come ulteriore preprocessing che le variabili vengano normalizzate in un range [0, 1]. Il parametro di tuning del modello è il numero di vicini ottimale per la *sensitivity*, in questo caso pari a 5 vicini.

Neural Network



Anche il neural network richiede come preprocessing la normalizzazione delle variabili in un range [0,1]. I parametri tunati sono:
size = numero di neuroni nascosti = 5;
decay = velocità di aggiornamento dei parametri = 0.01;
maxit = early stopping = 300 interazioni;

Ensamble



I modelli con sensitivity migliore sono stati usati per il modello ensemble: *neural network*, *random forest, bagging*,*gradient boosting* e *k-nearest neighbour*.

Stacking

Per lo stacking usiamo lo stesso sottoinsieme di modelli considerato per l'*ensemble*, ma questa volta viene coinvolto il modello logistico come Meta-Classifier.

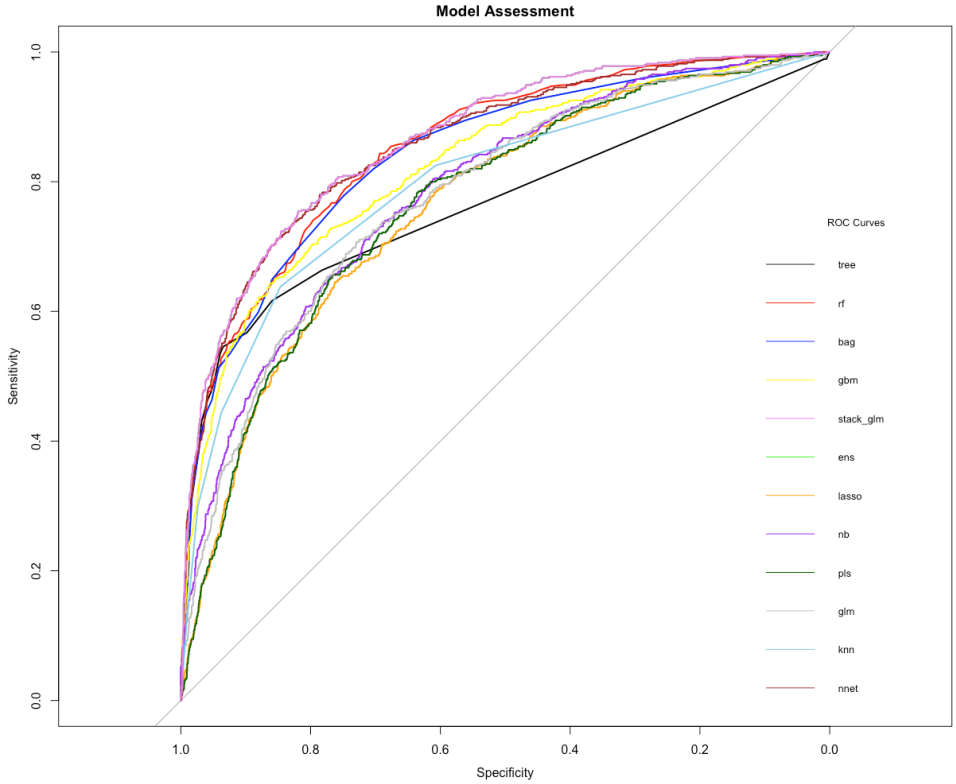
Valutazione overfitting

Confrontando le performance dei modelli su dataset di training e di validation è emerso che nessun modello overfitta (tutte le variazioni percentuali sono minori del 10%).

Modello	Performance (train)	Performance (test)	Scarto train - test
Tree	0.8600	0.8500	1.16%
Random forest	0.8578	0.8570	0.09%
Gradient boosting	0.8462	0.8474	-0.14%
Bagging	0.8502	0.8544	-0.49%
Logistico	0.8167	0.8155	0.15%
Lasso	0.8044	0.8136	-1.14%
PLS	0.8097	0.8018	0.98%
Naive Bayes	0.8189	0.8207	-0.22%
Knn	0.8432	0.8370	0.74%
Neural network	0.8607	0.8588	0.22%

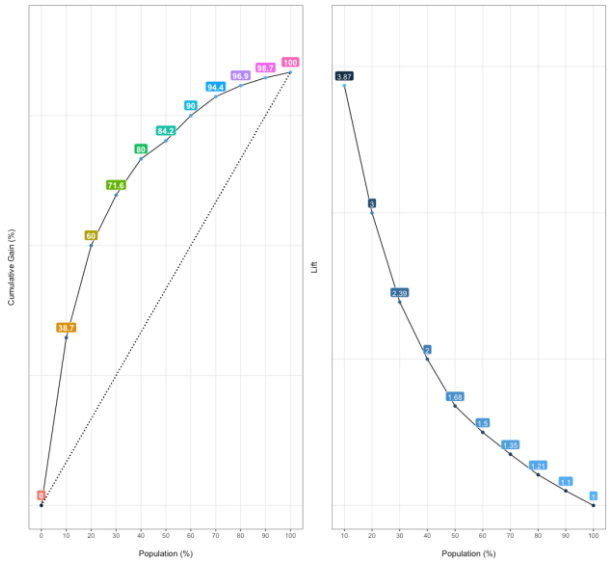
Step 2 - Assessment

Per scegliere il miglior modello tra i vari modelli dello step 1 si sono andate ad analizzare le curve ROC.

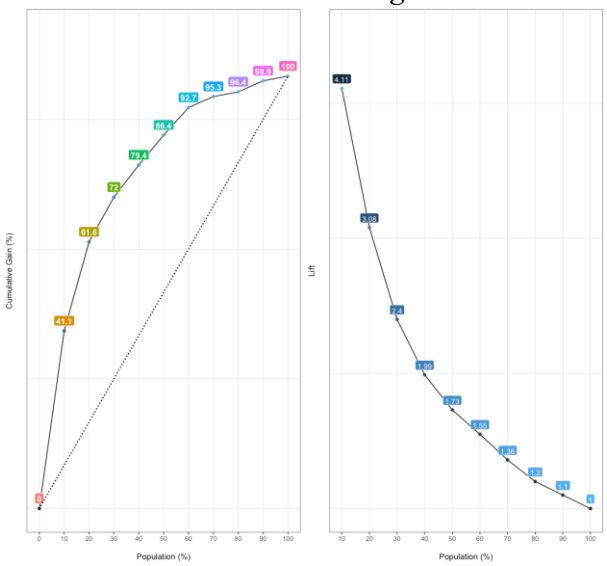


Dal grafico delle curve ROC si vede che il lasso, il PLS, il logistico, il tree e il naive bayes sono i metodi che forniscono i modelli classificativi peggiori. Mentre i modelli migliori sono random forest, stacking, glm, ensemble, neural network e k-nearest neighbor. Dato che queste curve ROC si intersecano non è possibile risalire ad un modello migliore su ogni soglia; analizziamo allora, attraverso le curve LIFT, quali sono i modelli che hanno migliori prestazioni sui primi quattro decili dei dati di validation. Siccome le curve LIFT di stacking e ensemble sono identiche (così come le loro curve ROC) si sceglie di mantenere solamente lo stacking nelle analisi successive.

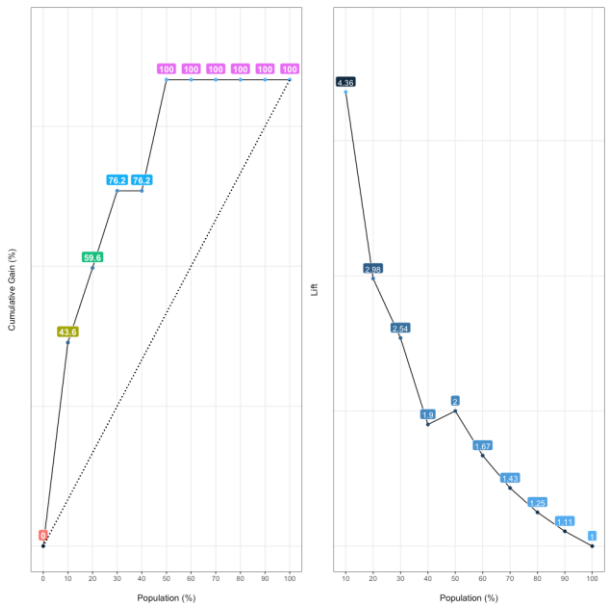
Lift di random forest



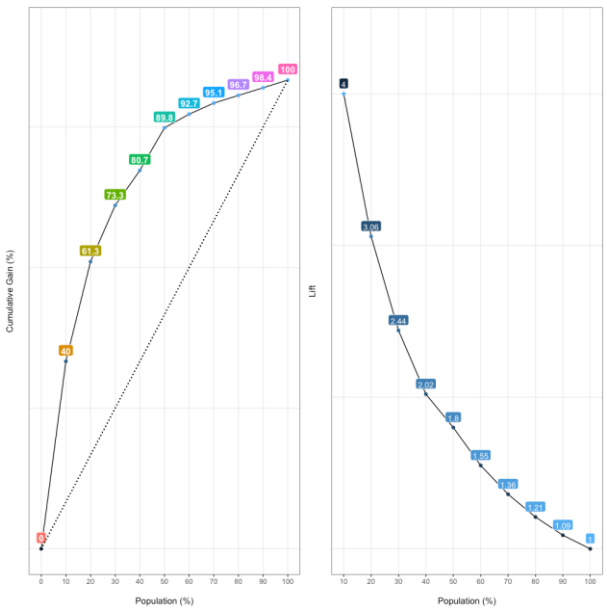
Lift di stack glm



Lift di knn



Lift di neural network

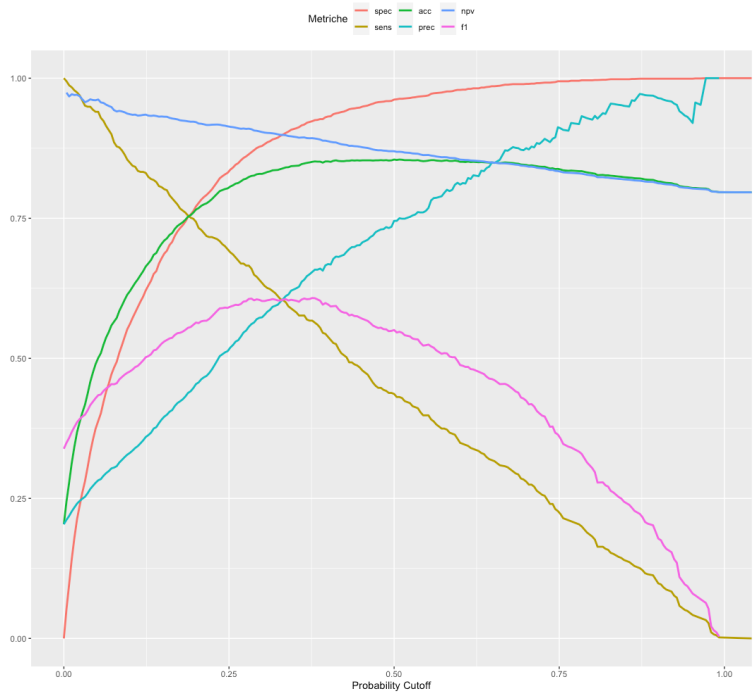


Nel primo 40% dei dati la *random forest* classifica correttamente l'80% dei *true churner*; viene dunque scelta come modello definitivo poiché, avendo prestazioni sui primi quattro decili pressoché pari a quelli della neural network, la r*andom forest* è un algoritmo che lascia più margine di interpretazione.

Step 3 - Model validation

Per scegliere la soglia è stato usato un criterio statistico: si osserva come nei dati di validazione variano diverse metriche. Sono state calcolate le metriche di *specificity, sensitivity, accuracy, precision, F1* e *negative predictive value*, e dato che la metrica di nostro interesse è la sensitivity, è stata scelta la soglia di 0.1.

Di seguito si riporta il grafico delle metriche al variare della soglia.



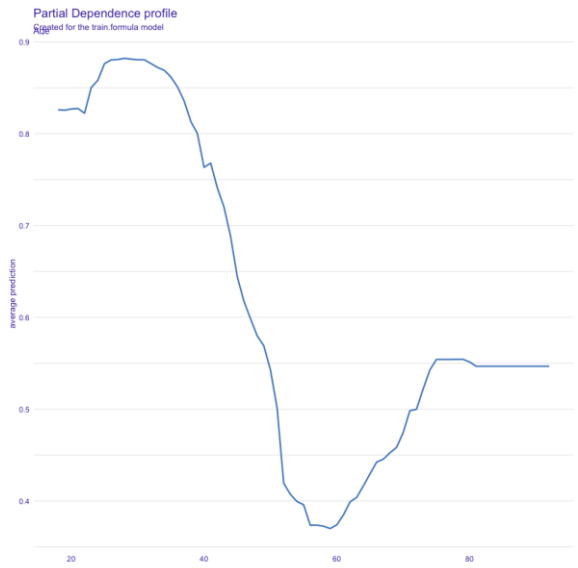
Il cut off pari a 0.1 significa che a una osservazione, affinché sia classificata come *churn*, deve essere assegnata da Random Forest una posterior superiore al suddetto valore.
Il fatto che una sufficiente una soglia così bassa è spiegata dalla rarità - relativa, si tratta comunque di un dataset 80/20 - dell'evento *churn*, nonché dalla "sporcizia" dei dati.

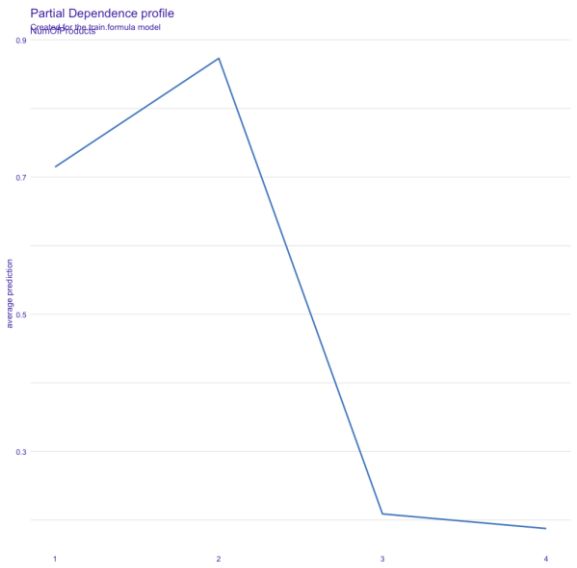
Step 4 - Score

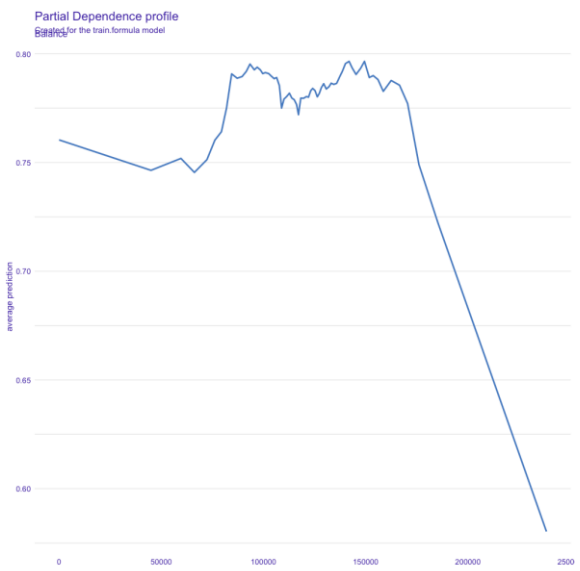
Dopo aver trovato la soglia adeguata, sono state calcolate, sui dati di score, le previsioni sui nuovi casi. La sensitivity è pari a 0.82.

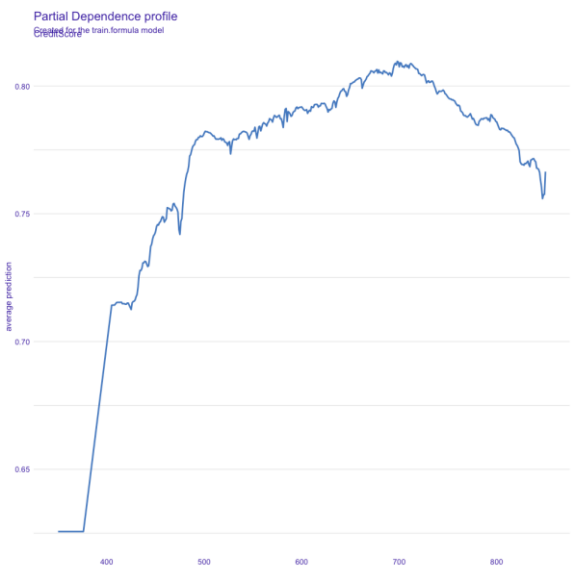
```
          Sensitivity : 0.8200
          Specificity : 0.6384
       Pos Pred Value : 0.3673
       Neg Pred Value : 0.9327
            Prevalence : 0.2038
       Detection Rate : 0.1671
 Detection Prevalence : 0.4550
    Balanced Accuracy : 0.7292
```

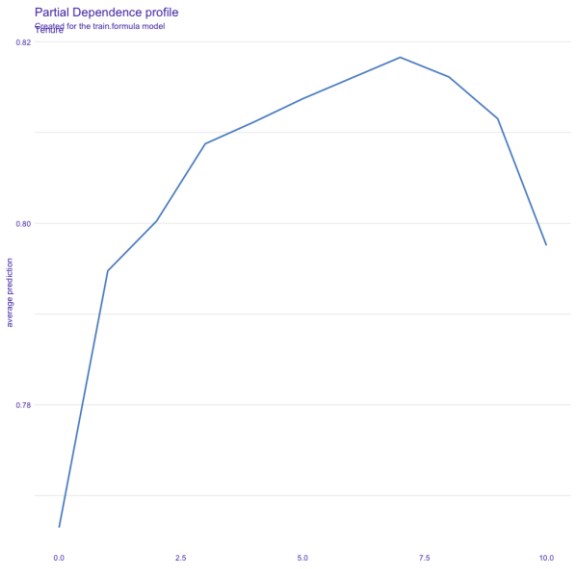
Interpretazione random forest



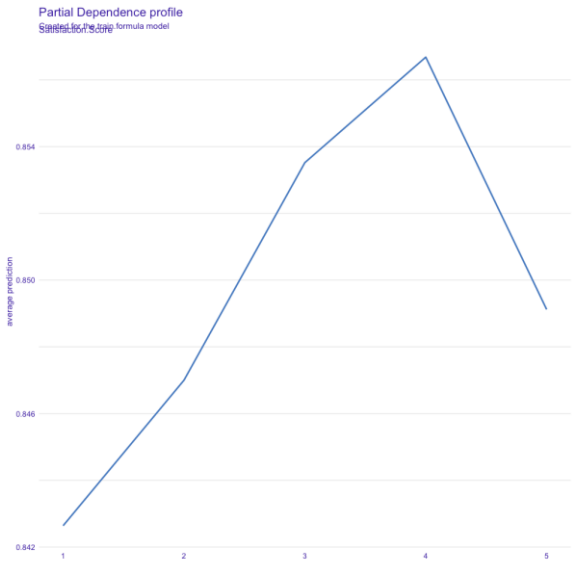


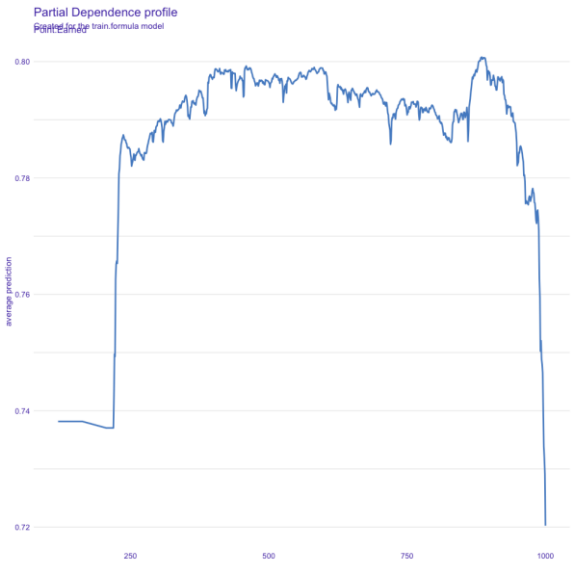

Partial Dependence profile
Created for the train.formula model
Balance
average prediction
0.80
0.75
0.70
0.65
0.60
0
50000
100000
150000
200000
2500


Partial Dependence profile
Created for the train.formula model
CreditScore
average prediction
0.80
0.75
0.70
0.65
400
500
600
700
800


Partial Dependence profile
Created for the train.formula model
Tenure
average prediction
0.82
0.80
0.78
0.0
2.5
5.0
7.5
10.0


Partial Dependence profile
Created for the train.formula model
EstimatedSalary
average prediction
0.82
0.80
0.78
0.76
0.74
0
50000
100000
150000
200000

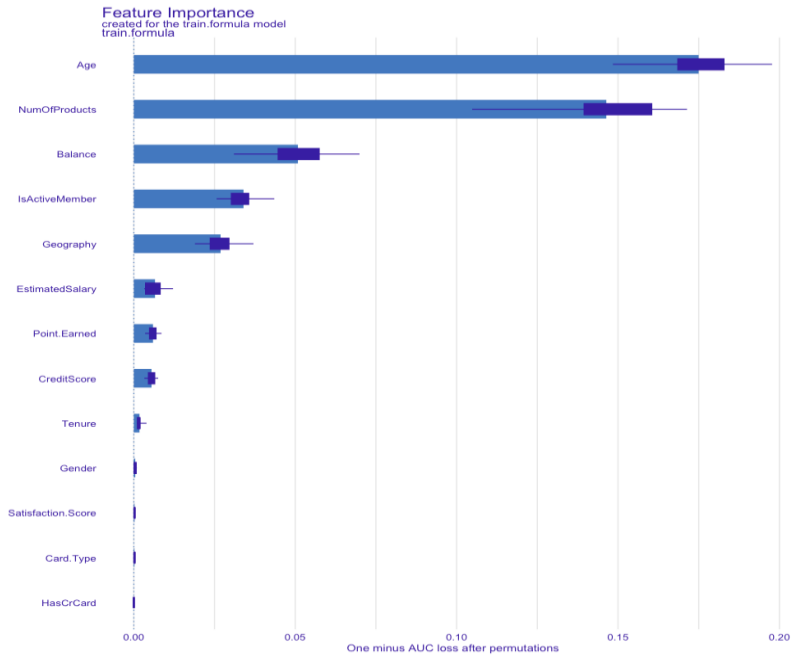

Partial Dependence profile
Created for the train.formula model
Satisfaction Score
average prediction
0.854
0.850
0.846
0.842
1
2
3
4
5


Partial Dependence profile
Created for the train.formula model
Point Earned
average prediction
0.80
0.78
0.76
0.74
0.72
250
500
750
1000

Nelle figure precedenti sono riportati i *Partial Dependence Plot* per le variabili numeriche, che evidenziano la posterior media attribuita da Random Forest al variare del valore di ciascuna variabile; sono quindi indicatrici del contributo della variabile sulla classificazione (di fatto, sulla posterior).
Si nota che:

- Il contributo di *Age* è rilevante e mostra una tendenza decrescente tra i 25 e i 60 anni, con inversione fino agli 80, con picco sui 25
- Il contributo di NumOfProdcuts è rilevante, con posterior media elevata fino a due prodotti e tendenza decrescente per numeri maggiori
- Il contributo di Balance è più ridotto (notare la scala) ma comunque presente; in particolare, bilanci elevanti mostrano andamento decrescente di probabilità di *churn*
- Il contributo di *CreditScore* è rilevante e si nota che a punteggi di merito di credito elevati corrispondono posterior di *churn* più elevate ma con andamento altalenante
- Il contributo delle rimanenti variabili sulle posterior medie calcolate da Random Forest è minore, per alcune di queste trascurabile.

Queste considerazioni sono concordi con le prossime figure:



Rappresentazione alternativa della *Feature Importance* per Random Forest in cui è presente un intervallo di confidenza per l'importanza di ciascuna variabile.

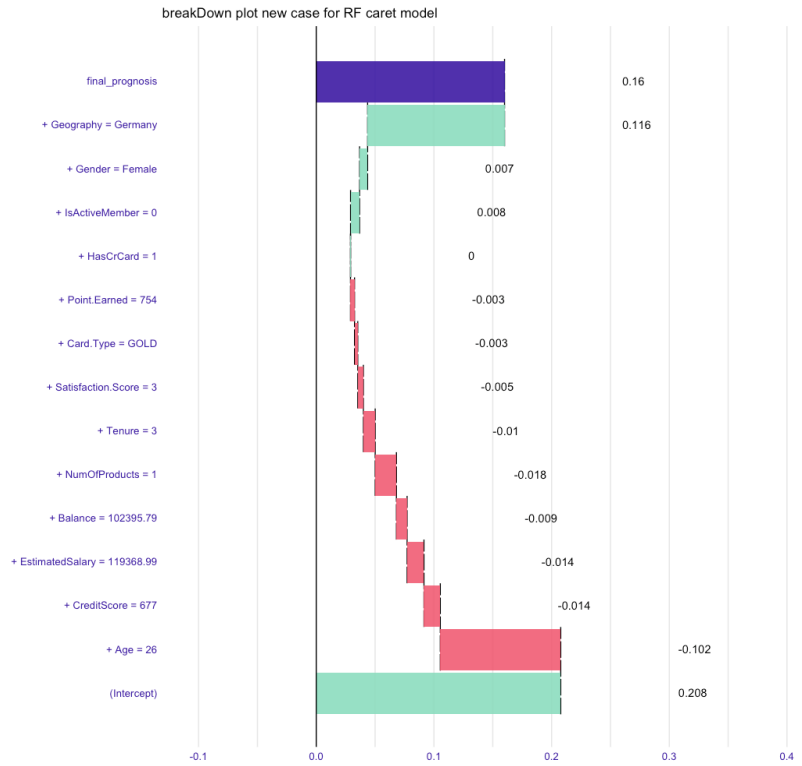


Figura relativa al contributo di ciascuna variabile, relativamente al valore rilevato sull'osservazione in questione, alla generazione della posterior su una osservazione presa a caso dal dataset di score.
Intercept si riferisce alla prior dell'evento nei dati, mentre *final_prognosis* riporta la posterior assegnata da Random Forest all'osservazione; il grafico mostra come il valore delle variabile influisce - anche numericamente - sul calcolo della posterior.

Riassunto concettuale dei passaggi effettuati