Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

Hotel Reservations Dataset

Obiettivo:

L'introduzione di nuove modalità di prenotazione hotel online ha cambiato profondamente l'approccio del cliente, molti annullano o non si presentano, talvolta incentivati dalla possibilità di cancellare gratuitamente o a basso costo. Questa *digital disruption* porta vantaggi al cliente ma un potenziale impatto negativo sulle entrate degli hotel.
Il nostro obbiettivo è quello di realizzare un modello classificativo capace di prevedere quali prenotazioni verranno presumibilmente cancellate.

Alleneremo il nostro modello sul dataset Hotel Reservation contenente informazioni riguardanti 36.238 prenotazioni provenienti dalla catena alberghiera americana *Inn*.

Analisi preliminare e nuove variabili:

Il dataset contiene informazioni generali sulla prenotazione come il numero di ospiti (***no_of_adults, no_of_children***), il periodo e il preavviso della prenotazione (***no_of_weekend_nights, no_of_week_nights, no_of_nights, arrival_year, arrival_month, arrival_day, lead_time***) e il prezzo (***avg_price_per_room***). Sono presenti, inoltre, dettagli sul trattamento come pasti, parcheggio, tipo di stanza, altre eventuali richieste (***type_of_meal_plan, required_car_parking_space, room_type_reserved, no_of_special_requests***) e dettagli sul cliente (***market_segment_type, repeated_guest, no_of_previous_cancellations, no_of_previous_booking_not_cancelled***).
Decidiamo inoltre di ricavare una variabile riepilogativa ***arrival*** in formato data aaaa-mm-dd composta dalle tre variabili precedentemente citate. Fatto ciò, possiamo ricavare il giorno della settimana previsto per l'arrivo (***arrival_weekday***), utile per un'analisi iniziale.

Il target (***booking_status***) è di tipo binario e riporta lo stato della prenotazione: Canceled o Not_Canceled. Osserviamo come si distribuisce:

```
> prop.table(table(data1$booking_status))

  Canceled Not_Canceled
 0.3277775    0.6722225
```

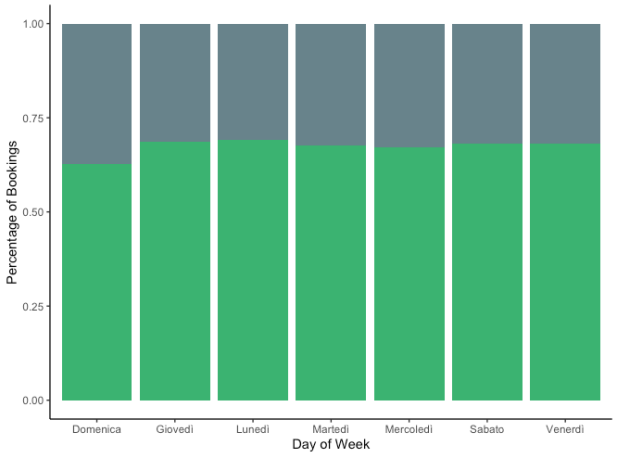
Tramite una rapida ricerca possiamo dire che la percentuale di prenotazioni cancellate è elevata ma realistica, soprattutto se la catena di hotel in questione offre una politica di cancellazione flessibile.

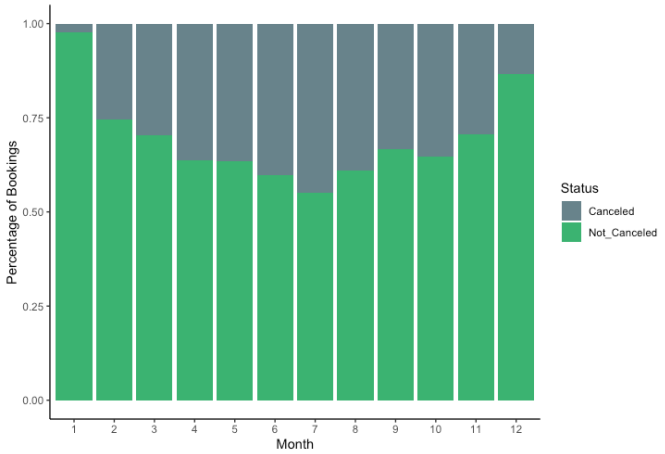
Dato che disponiamo di un solo set di dati, è nostra premura suddividerlo in train, validation e score accertandoci che la distribuzione del target resti invariata. Prima di procedere nella suddivisione (e riduzione viste le elevate dimensioni), controlliamo la presenza di valori mancanti e univoci. Il dataset non presenta dati mancanti, notiamo inoltre che la variabile ***Booking_ID*** è univoca per ogni osservazione, decidiamo quindi di rimuoverla.

```
> status
                              variable q_zeros p_zeros q_na p_na q_inf p_inf      type unique
1                           Booking_ID       0    0.00    0    0     0     0 character  36238
2                         no_of_adults     139    0.38    0    0     0     0   integer      5
3                        no_of_children   33544   92.57    0    0     0     0   integer      6
4                  no_of_weekend_nights   16872   46.56    0    0     0     0   integer      8
5                     no_of_week_nights    2383    6.58    0    0     0     0   integer     18
6                     type_of_meal_plan       0    0.00    0    0     0     0    factor      4
7               required_car_parking_space   35117   96.91    0    0     0     0    factor      2
8                   room_type_reserved       0    0.00    0    0     0     0    factor      7
9                            lead_time    1295    3.57    0    0     0     0   integer    352
10                        arrival_year       0    0.00    0    0     0     0   integer      2
11                       arrival_month       0    0.00    0    0     0     0    factor     12
12                        arrival_date       0    0.00    0    0     0     0   integer     31
13                  market_segment_type       0    0.00    0    0     0     0    factor      5
14                      repeated_guest   35312   97.44    0    0     0     0    factor      2
15          no_of_previous_cancellations   35901   99.07    0    0     0     0   integer      9
16 no_of_previous_bookings_not_canceled   35429   97.77    0    0     0     0   integer     59
17                    avg_price_per_room     545    1.50    0    0     0     0   numeric   3919
18                no_of_special_requests   19751   54.50    0    0     0     0   integer      6
19                        booking_status       0    0.00    0    0     0     0    factor      2
20                          no_of_nights      78    0.22    0    0     0     0   integer     25
21                               arrival       0    0.00    0    0     0     0      Date    549
22                       arrival_weekday       0    0.00    0    0     0     0    factor      7
```

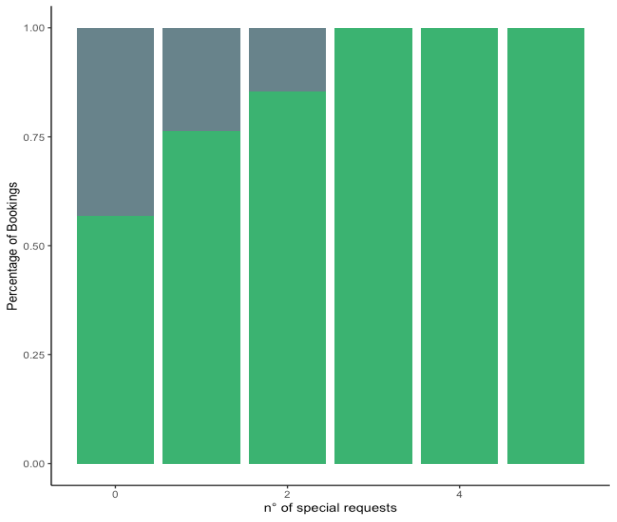
Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

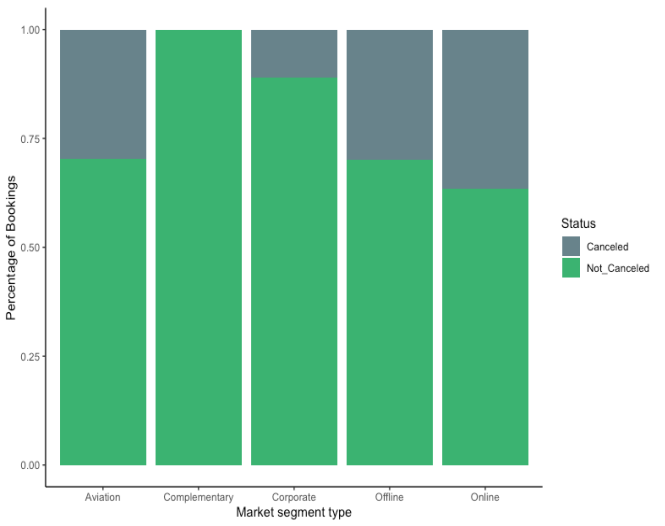
Tramite il pacchetto caret otteniamo tre partizioni rispettivamente di 6847 (train), 2937 (validation), 1088 (score) osservazioni. Nel dataset di score cancelliamo la colonna contenente il target.
Osservando la distribuzione delle variabili categoriali rispetto al target confermiamo che non ci sono "target nascosti", non notiamo quindi problemi di separation che renderebbero inutilizzabili i modelli. Vediamo ora come si distribuisce il target rispetto alle variabili temporali degne di nota.





Si può dedurre che il giorno di arrivo non influenzi particolarmente l'esito della prenotazione. Al contrario, dividendo le osservazioni rispetto al mese di soggiorno, si nota come nei mesi estivi la frequenza delle cancellazioni è molto elevata, raggiungendo una soglia poco inferiore al 50% nel mese di luglio. Possiamo interpretare il fenomeno come impulsività da parte dei clienti nella prenotazione di un alloggio per le vacanze estive. Nei mesi freddi il fenomeno è molto ridotto: nel mese di gennaio la frequenza delle cancellazioni è inferiore al 10%.



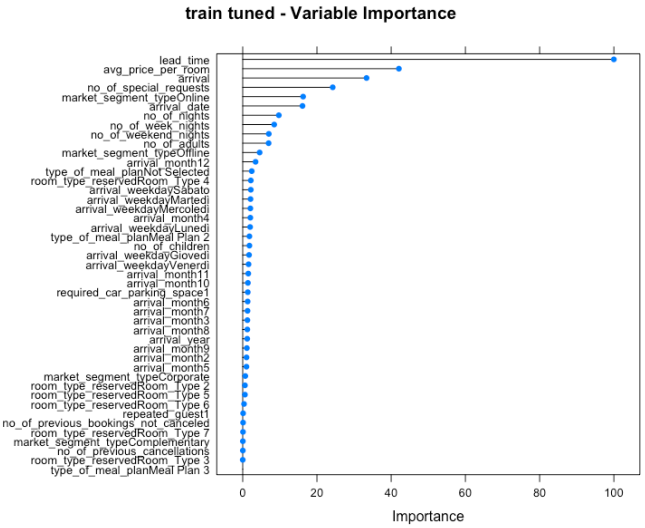


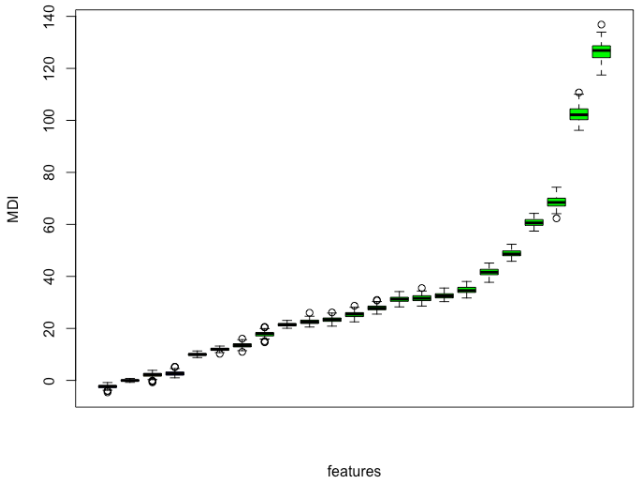
Altre variabili che portano a conclusioni interessanti sono il numero di richieste speciali eseguite e la designazione del segmento di mercato. I clienti esigenti, ovvero coloro che formulano oltre tre richieste speciali al momento della prenotazione, non disdicono mai. Lo stesso vale per i clienti che prenotano tramite servizi *"complementary"*, come potrebbero essere pacchetti viaggio. Anche i clienti che prenotano per motivi aziendali (*corporate*) sono poco propensi a cancellare il soggiorno.

Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

Model selection:

Per eseguire model selection utilizziamo inizialmente un modello random forest, quest'ultimo ci porterebbe a tenere tutte le variabili. Per avere un secondo parere utilizziamo anche l'algoritmo Boruta, il quale opera creando associazioni casuali tra le variabili e confrontandone le prestazioni con le features originali, terminate le combinazioni, risulta che la variabile ***no_of_previous_cancellations*** sia poco rilevante. Decidiamo quindi di rimuoverla.





Modelli di classificazione:

La Sensitivity misura la capacità del modello di identificare correttamente le prenotazioni che vengono cancellate e rappresenta la percentuale di veri positivi correttamente identificati. Se la priorità è identificare correttamente le prenotazioni cancellate, la Sensitivity è una buona metrica da utilizzare.
Tuttavia, la ROC fornisce un'immagine completa della performance del modello in termini di capacità di rilevare veri positivi, anche a fronte di un aumento del numero di falsi positivi. La ROC è generalmente considerata migliore rispetto ad altre metriche come Sensitivity o Specificity, perché tiene conto della relazione tra TPR e FPR, e quindi consente una valutazione più completa della performance del modello.
Decidiamo quindi di utilizzare la metrica ROC nei modelli classificativi, dando tuttavia maggior importanza alla sensitivity nel momento della scelta della soglia ottimale.

Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

Generalized linear model

Validation:

```
Confusion Matrix and Statistics

             Reference
Prediction     Canceled Not_Canceled
  Canceled          605          196
  Not_Canceled      358         1778

               Accuracy : 0.8114
                 95% CI : (0.7967, 0.8254)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5528

 Mcnemar's Test P-Value : 7.906e-12

            Sensitivity : 0.6282
            Specificity : 0.9007
         Pos Pred Value : 0.7553
         Neg Pred Value : 0.8324
             Prevalence : 0.3279
         Detection Rate : 0.2060
   Detection Prevalence : 0.2727
      Balanced Accuracy : 0.7645
```

Tunato utilizzando il dataset sottoposto a model selection. I parametri "corr" e "nzv" selezionati nel preprocessing risolvono rispettivamente i problemi di collinearità e di near zero variance presenti nei dati.

Train confusion matrix:

```
             Reference
Prediction     Canceled Not_Canceled
  Canceled         20.2          8.1
  Not_Canceled     12.1         59.6

 Accuracy (average) : 0.7979
```

Lasso

Validation:

```
Confusion Matrix and Statistics

             Reference
Prediction     Canceled Not_Canceled
  Canceled          592          177
  Not_Canceled      371         1797

               Accuracy : 0.8134
                 95% CI : (0.7988, 0.8274)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5536

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6147
            Specificity : 0.9103
         Pos Pred Value : 0.7698
         Neg Pred Value : 0.8289
             Prevalence : 0.3279
         Detection Rate : 0.2016
   Detection Prevalence : 0.2618
      Balanced Accuracy : 0.7625

       'Positive' Class : Canceled
```

La regressione Lasso è un metodo di regressione che limita la quantità di coefficienti utilizzati nella previsione per prevenire overfitting.
Richiede lo stesso preprocessing del modello logistico ma riesce a lavorare con il dataset antecedente alla model selection.

Train confusion matrix:

```
             Reference
Prediction     Canceled Not_Canceled
  Canceled         19.5          7.0
  Not_Canceled     13.4         60.0

 Accuracy (average) : 0.7955
```

Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

Neural network

Validation:

```
Confusion Matrix and Statistics

             Reference
Prediction     Canceled Not_Canceled
  Canceled          673          217
  Not_Canceled      290         1757

               Accuracy : 0.8274
                 95% CI : (0.8132, 0.8409)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6006

 Mcnemar's Test P-Value : 0.001386

            Sensitivity : 0.6989
            Specificity : 0.8901
         Pos Pred Value : 0.7562
         Neg Pred Value : 0.8583
             Prevalence : 0.3279
         Detection Rate : 0.2291
   Detection Prevalence : 0.3030
      Balanced Accuracy : 0.7945

       'Positive' Class : Canceled
```

La rete neurale tunata è composta da quattro livelli nascosti e un livello di output. Richiede tutto il preprocessing del logistico più un'ulteriore normalizzazione degli input per migliorare la convergenza.

Train confusion matrix:

```
             Reference
Prediction     Canceled Not_Canceled
  Canceled         22.0          8.4
  Not_Canceled     10.7         59.0

 Accuracy (average) : 0.8094
```

Tree

Validation:

```
Confusion Matrix and Statistics

             Reference
Prediction     Canceled Not_Canceled
  Canceled          597          107
  Not_Canceled      366         1867

               Accuracy : 0.839
                 95% CI : (0.8252, 0.8521)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6076

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6199
            Specificity : 0.9458
         Pos Pred Value : 0.8480
         Neg Pred Value : 0.8361
             Prevalence : 0.3279
         Detection Rate : 0.2033
   Detection Prevalence : 0.2397
      Balanced Accuracy : 0.7829

       'Positive' Class : Canceled
```

L'albero decisionale non necessita di preprocessing, il criterio di split e la regola di arresto sono gestite dal pacchetto Caret.

Train confusion matrix:

```
             Reference
Prediction     Canceled Not_Canceled
  Canceled         21.8          6.1
  Not_Canceled     10.9         61.2

 Accuracy (average) : 0.83
```

Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

Gradient boosting

Validation:

```
Confusion Matrix and Statistics

              Reference
Prediction     Canceled Not_Canceled
  Canceled          724          150
  Not_Canceled      239         1824

               Accuracy : 0.8676
                 95% CI : (0.8548, 0.8796)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6922

 Mcnemar's Test P-Value : 8.128e-06

            Sensitivity : 0.7518
            Specificity : 0.9240
         Pos Pred Value : 0.8284
         Neg Pred Value : 0.8841
             Prevalence : 0.3279
         Detection Rate : 0.2465
   Detection Prevalence : 0.2976
      Balanced Accuracy : 0.8379

       'Positive' Class : Canceled
```

Questo algoritmo basato sugli alberi opera minimizzando la devianza ad ogni iterazione. Essendo un modello ensamble non richiede preprocessing.

Train confusion matrix:

```
              Reference
Prediction     Canceled Not_Canceled
  Canceled         23.7          5.4
  Not_Canceled      9.2         61.8

 Accuracy (average) : 0.8545
```

Bagging Trees

Validation:

```
Confusion Matrix and Statistics

              Reference
Prediction     Canceled Not_Canceled
  Canceled          757          158
  Not_Canceled      206         1816

               Accuracy : 0.8761
                 95% CI : (0.8636, 0.8878)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.7152

 Mcnemar's Test P-Value : 0.01376

            Sensitivity : 0.7861
            Specificity : 0.9200
         Pos Pred Value : 0.8273
         Neg Pred Value : 0.8981
             Prevalence : 0.3279
         Detection Rate : 0.2577
   Detection Prevalence : 0.3115
      Balanced Accuracy : 0.8530

       'Positive' Class : Canceled
```

Ulteriore algoritmo di ensamble, lavora con 250 alberi e ad ogni iterazione i dati vengono perturbati per aumentarne la variabilità.

Train confusion matrix:

```
              Reference
Prediction     Canceled Not_Canceled
  Canceled         24.3          5.8
  Not_Canceled      8.4         61.4

 Accuracy (average) : 0.8573
```

Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

Glm stacking

Validation:

```
Confusion Matrix and Statistics

             Reference
Prediction     Canceled Not_Canceled
  Canceled          758          109
  Not_Canceled      205         1865

               Accuracy : 0.8931
                 95% CI : (0.8813, 0.904)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7511

 Mcnemar's Test P-Value : 8.269e-08

            Sensitivity : 0.7871
            Specificity : 0.9448
         Pos Pred Value : 0.8743
         Neg Pred Value : 0.9010
             Prevalence : 0.3279
         Detection Rate : 0.2581
   Detection Prevalence : 0.2952
      Balanced Accuracy : 0.8660

       'Positive' Class : Canceled
```

Questo algoritmo ensamble utilizza un modello logistico come meta-classificatore. I modelli utilizzati per le prediction sono: glm, lasso, knn, tree, pls, naive bayes, bagging, gb, rf, e nn.

Coefficients:

```
        glm       rpart         knn      glmnet         pls
-5.38094762  0.52667728 -0.51790379  3.97209871  0.32203748

 naive_bayes         gbm          rf        nnet     treebag
-0.06495666 -1.27510209 -4.06341637  0.47430022 -0.59535639
```

Random forest

Validation:

```
Confusion Matrix and Statistics

             Reference
Prediction     Canceled Not_Canceled
  Canceled          756          106
  Not_Canceled      207         1868

               Accuracy : 0.8934
                 95% CI : (0.8817, 0.9044)
    No Information Rate : 0.6721
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7515

 Mcnemar's Test P-Value : 1.583e-08

            Sensitivity : 0.7850
            Specificity : 0.9463
         Pos Pred Value : 0.8770
         Neg Pred Value : 0.9002
             Prevalence : 0.3279
         Detection Rate : 0.2574
   Detection Prevalence : 0.2935
      Balanced Accuracy : 0.8657

       'Positive' Class : Canceled
```

Simile al bagging, la differenza è che ad ogni interazione vengono creati alberi utilizzando un sottinsieme casuale di predittori del campione bootstrap estratto. La classificazione è poi decretata da un majority vote.

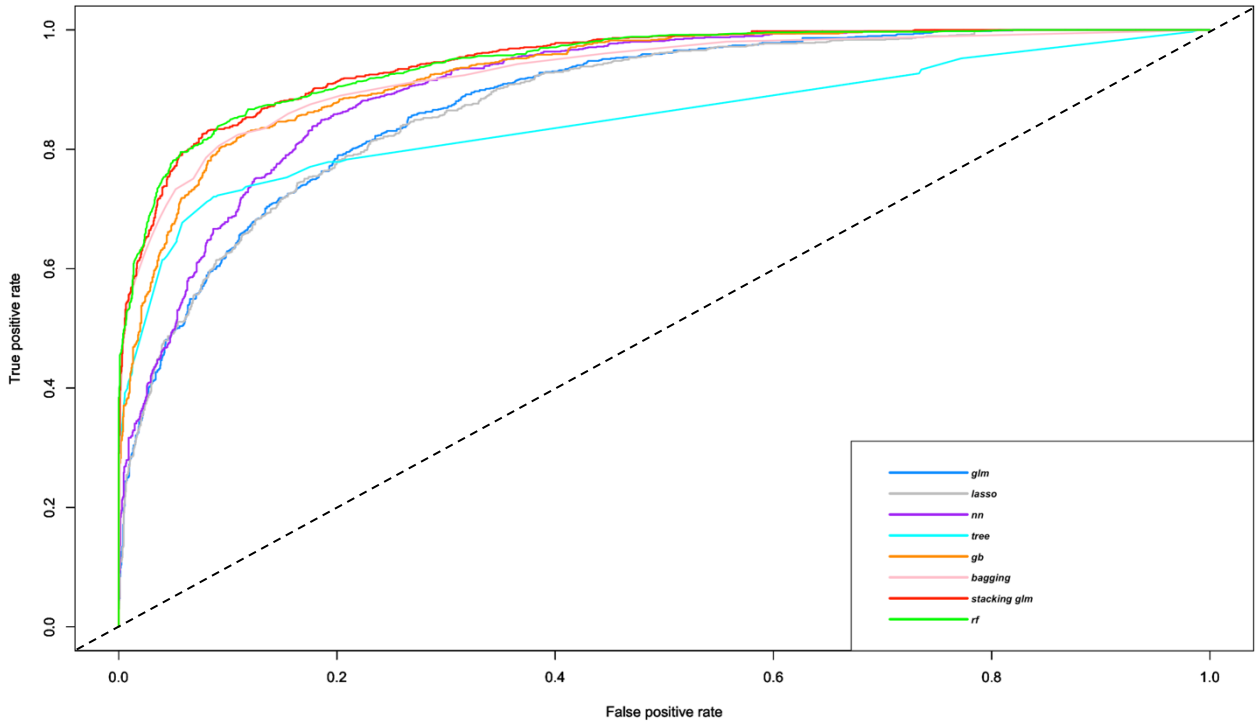
Train confusion matrix:

```
             Reference
Prediction     Canceled Not_Canceled
  Canceled         24.4          4.9
  Not_Canceled      8.3         62.4

 Accuracy (average) : 0.868
```

Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

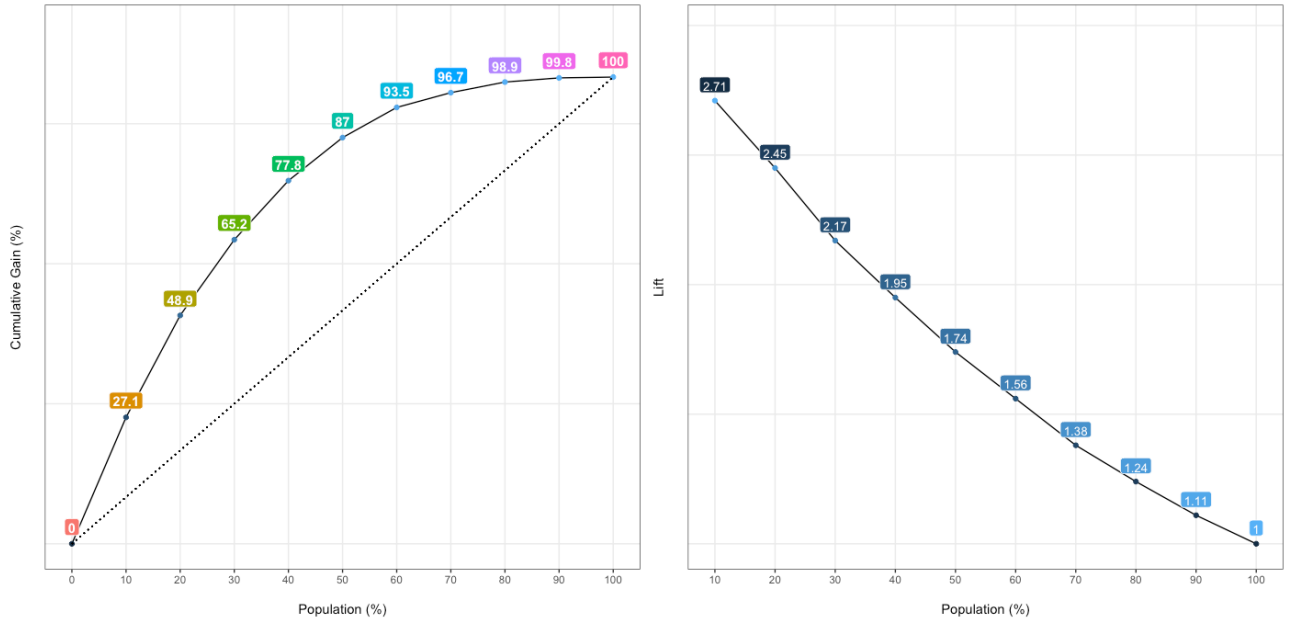
Selezione del modello ottimale:

Curve ROC:



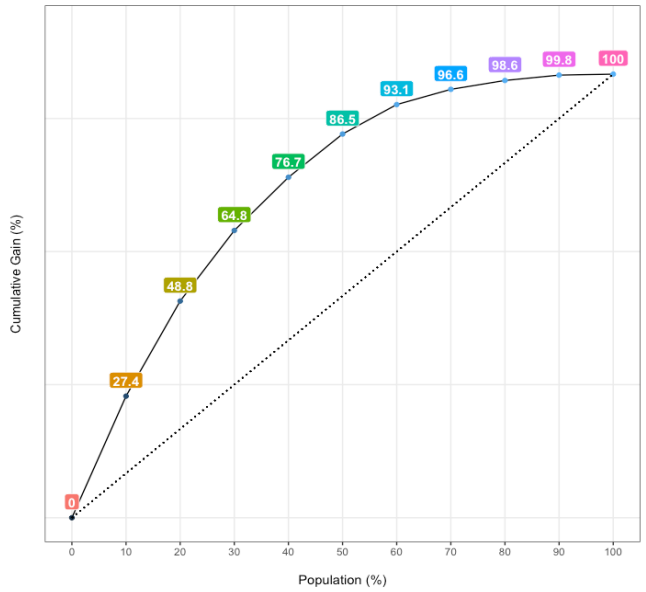
Tutti i modelli osservati performano bene sul dataset di validation, gli algoritmi che classificano peggio sono il Gerenalized linear model e la Lasso regression. Notiamo inoltre una curva particolare generata dall'albero decisionale, cresce rapidamente ma poi si assesta a un certo valore di TPR e la salita prosegue lenta, il modello sembra quindi avere difficoltà a differenziare i veri e i falsi positivi.
I modelli che performano meglio sono la Random forest, lo Stacking glm e il Bagging, la curva ROC non ci fornisce un vincitore evidente quindi osserviamo le curve lift:

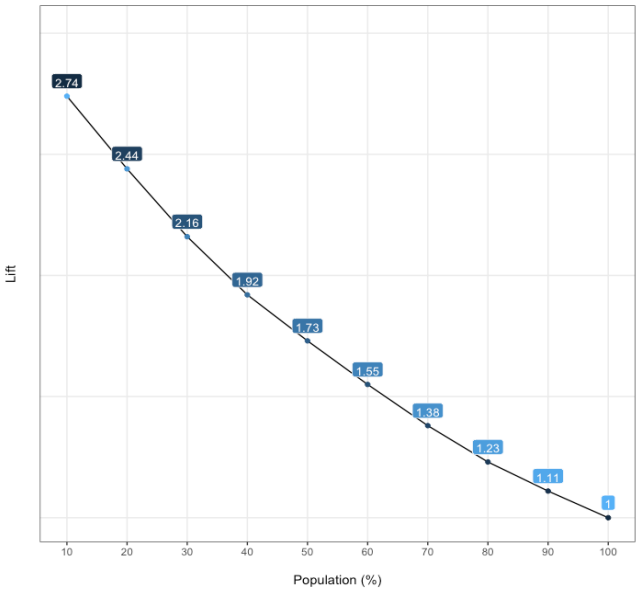
Glm:



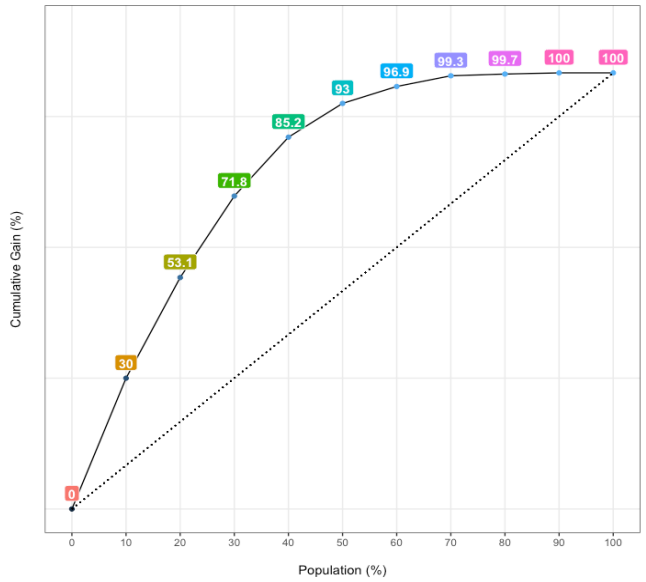
Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

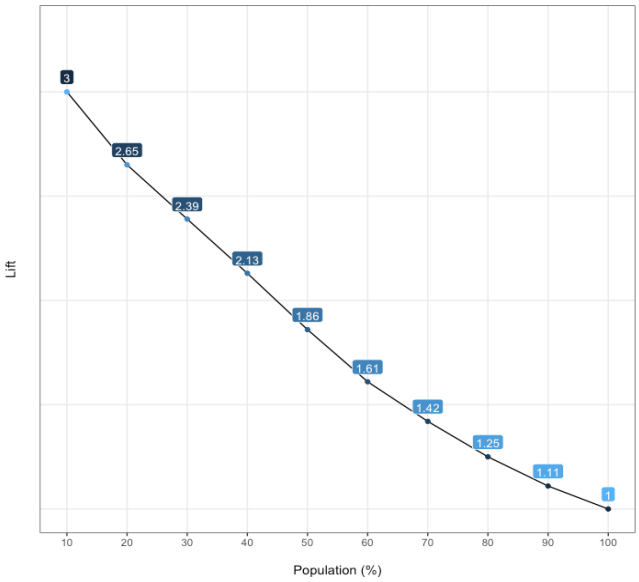
Lasso:



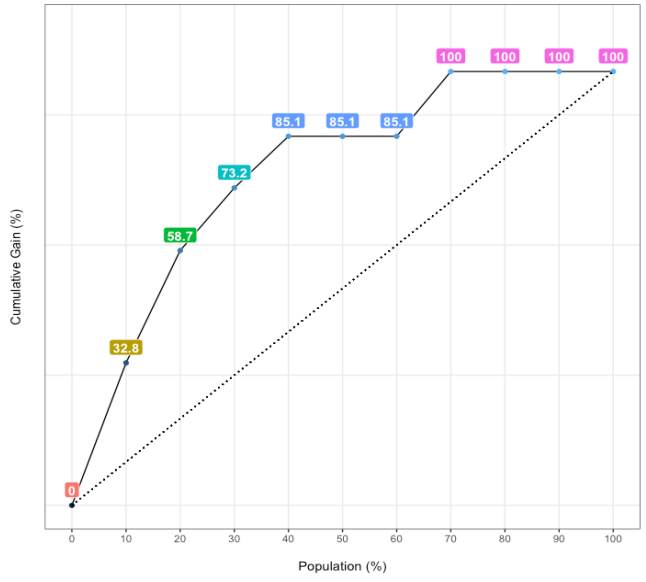


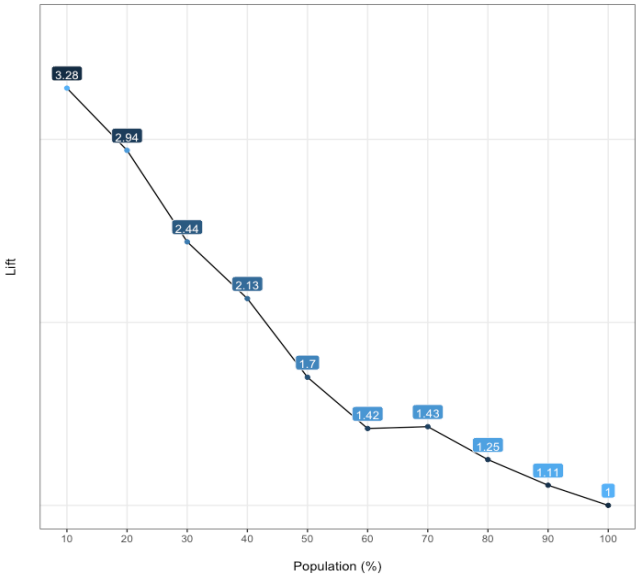
Neural network:





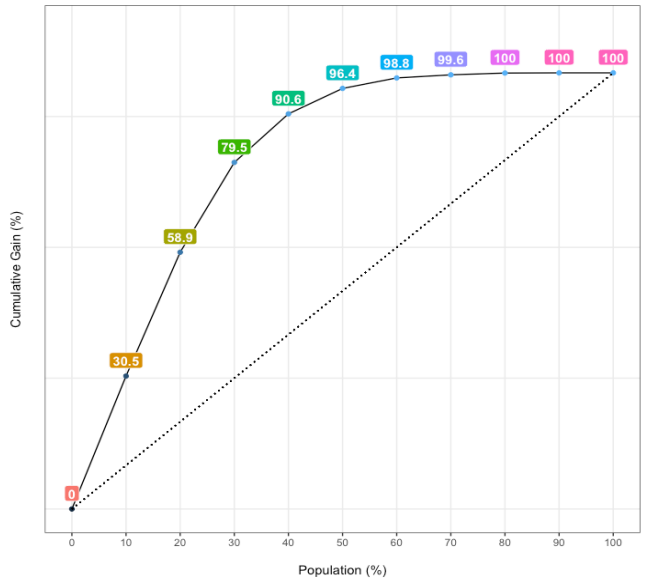
Tree:

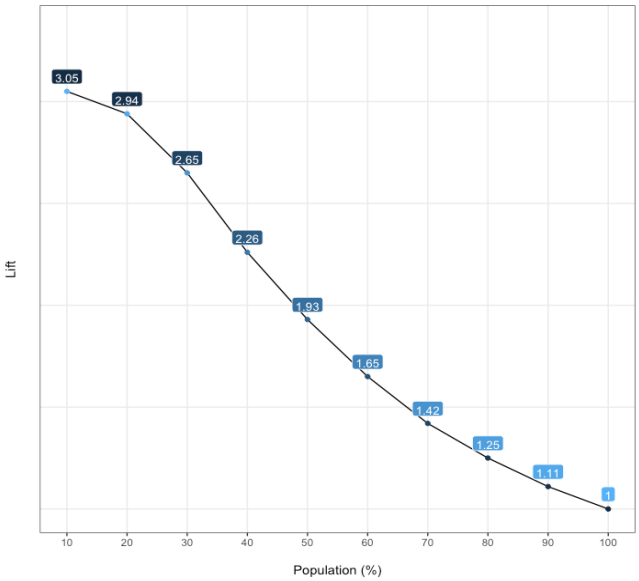




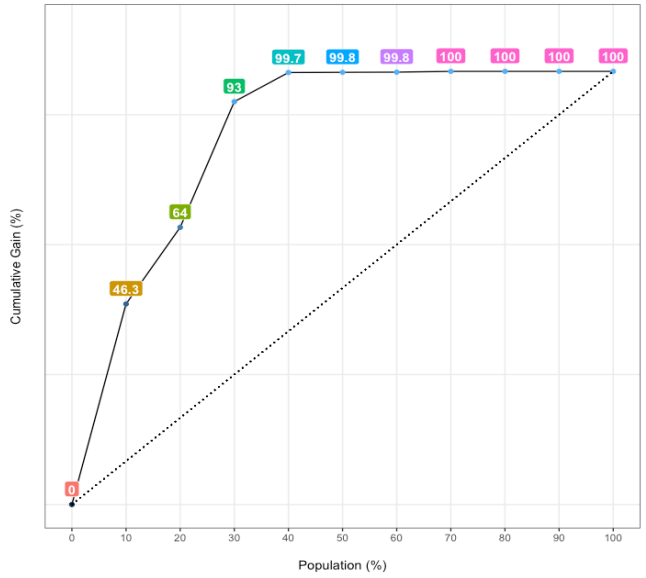
Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

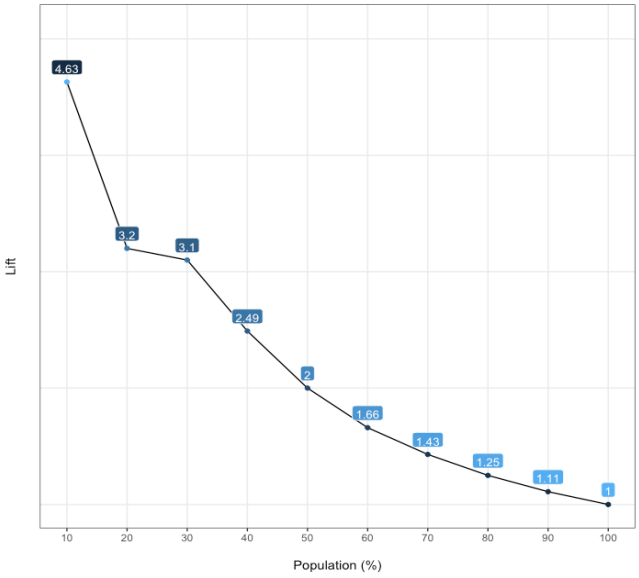
Gradient boosting:



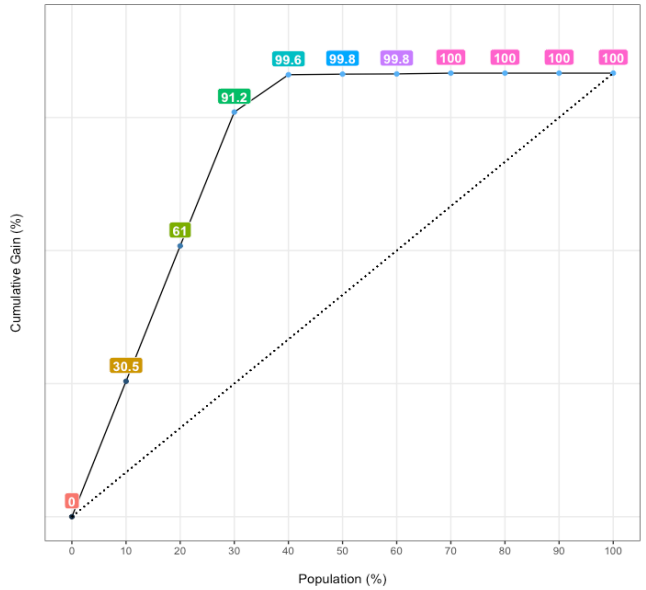


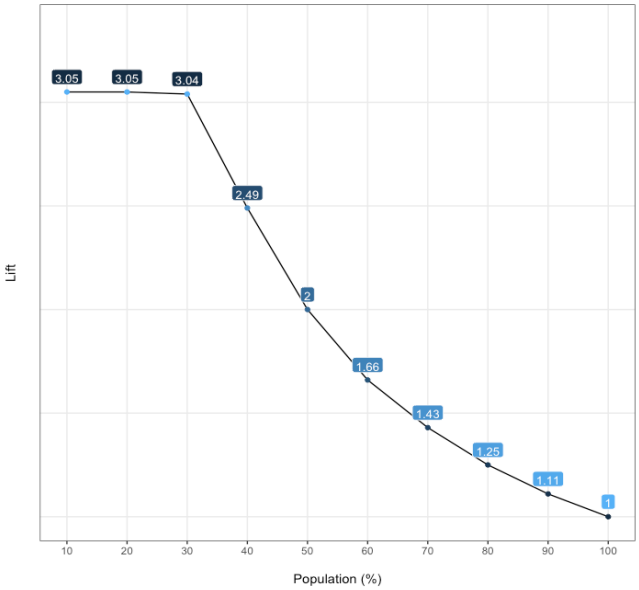
Bagging:



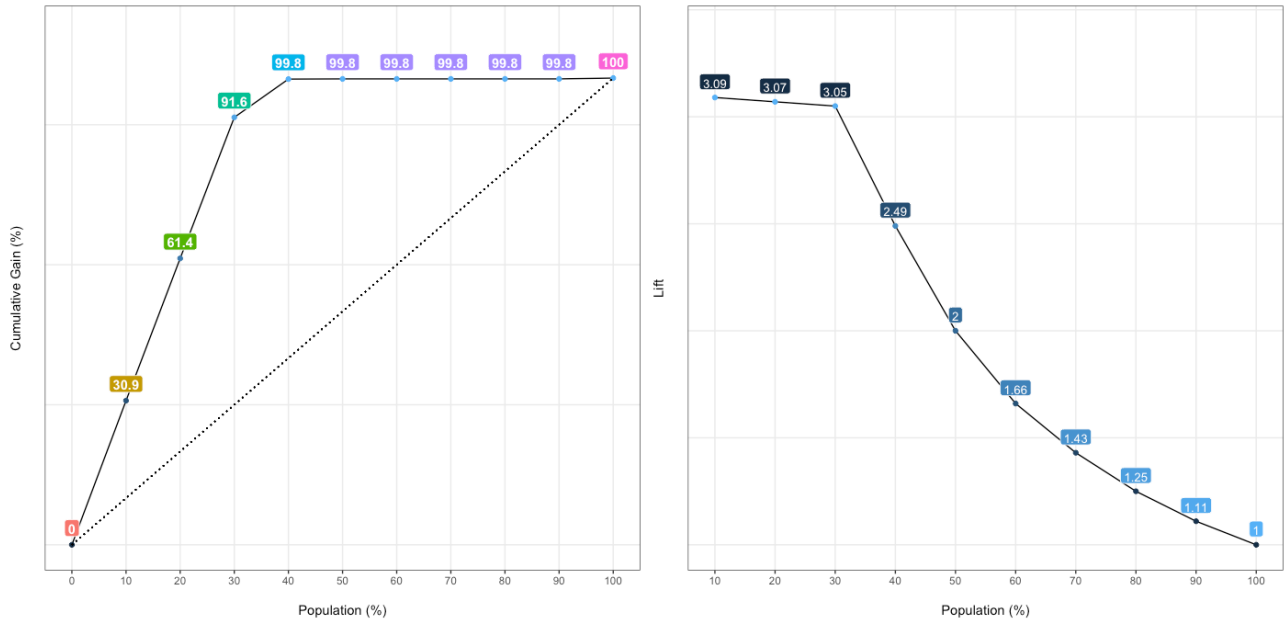


Stacking glm:



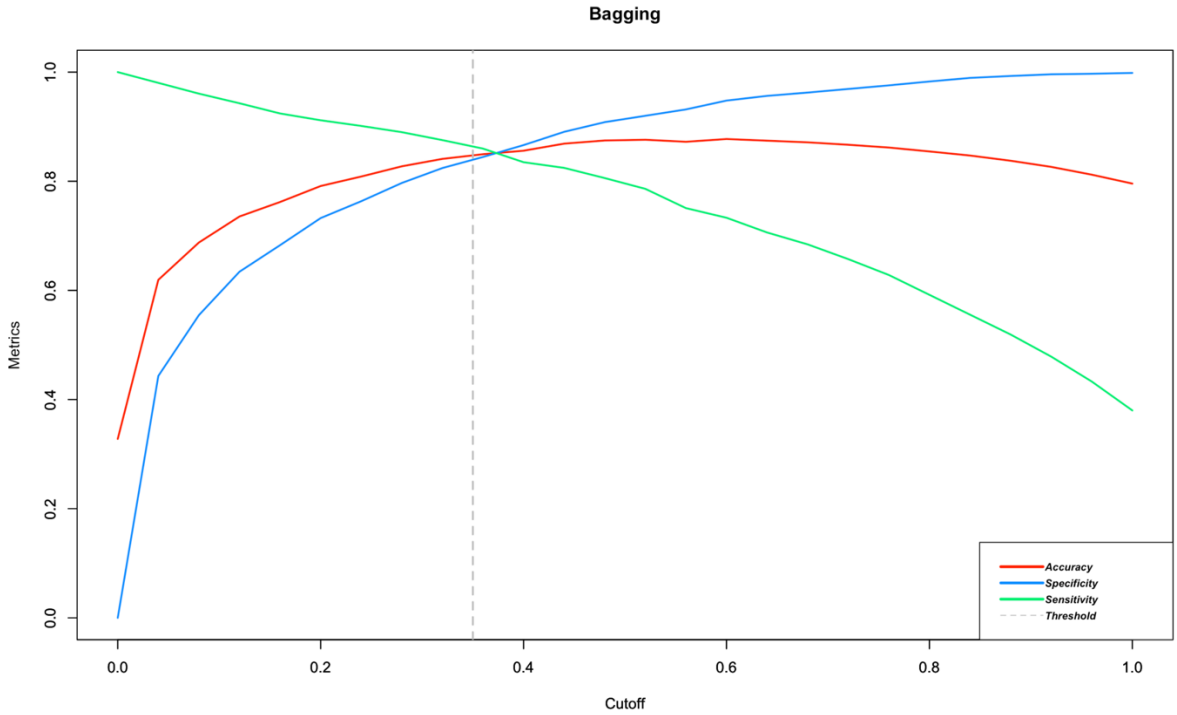


Random forest:



Nonostante osservando la curva ROC sembrasse che il miglior modello fosse conteso tra Random forest e Stacking glm, le curve Lift ci portano a scegliere il Bagging come classificatore. Infatti, considerando il 20% delle osservazioni con la prediction più alta, il modello riesce a catturare il 64% delle prenotazioni cancellate.

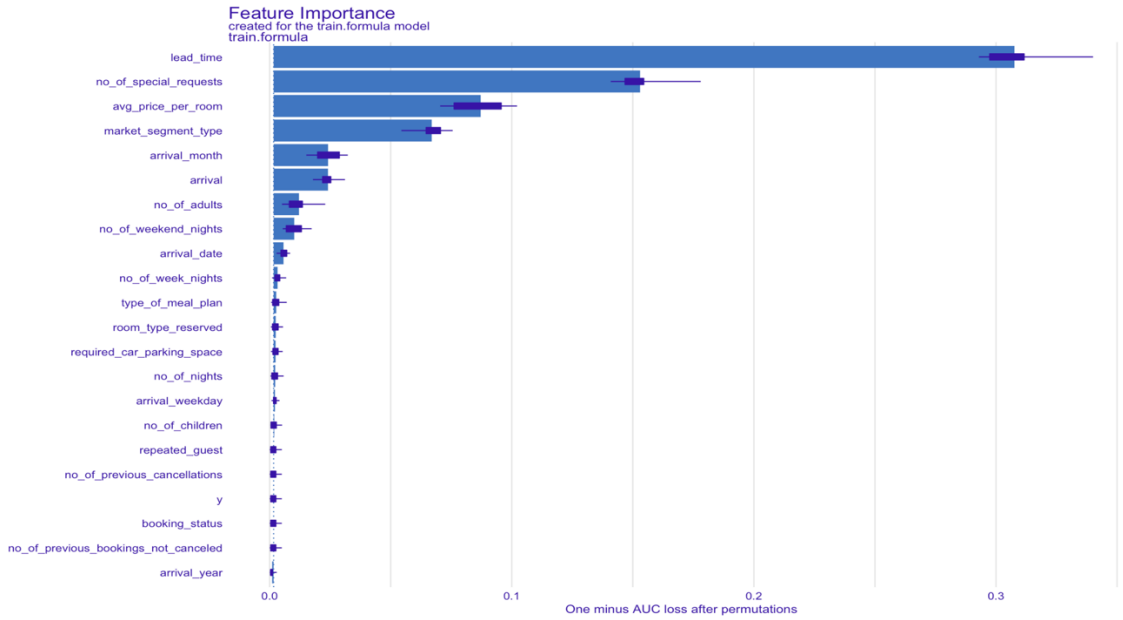
Selezione della soglia:



La Sensitivity misura la capacità del modello di identificare correttamente le prenotazioni che vengono cancellate, è quindi la metrica più di riguardo rispetto al nostro target. Per questo motivo scegliamo come soglia 0.35, trade-off in cui la sensitivity è abbastanza elevata, senza andare troppo a discapito di specificity e accuracy.

Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

Analisi delle variabili più importanti:



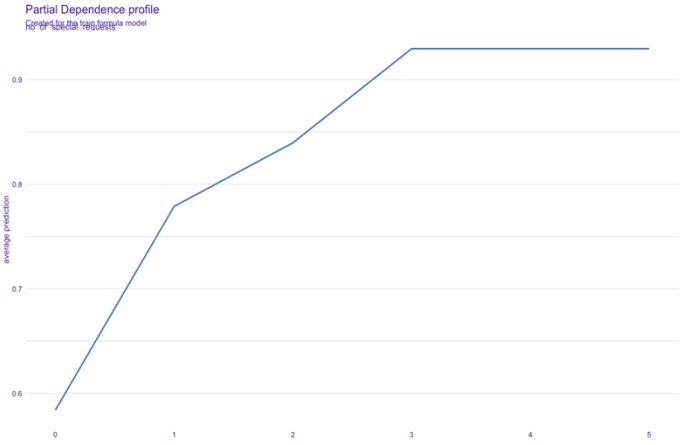
Il tempo di preavviso della prenotazione è nettamente la variabile più importante, seguito dal numero di richieste speciali eseguite in fase di prenotazione e dal prezzo medio per stanza. Dato che questa rappresentazione non ci fornisce informazioni sulla direzione in cui operano le variabili, osserviamo le dipendenze parziali tra la prediction e le tre variabili più importanti:

Lead time:

Maggiore è il tempo di preavviso al momento della prenotazione, più è probabile che la prenotazione verrà cancellata. Osserviamo che mediamente il modello classifica le osservazioni come "canceled" se la prenotazione viene effettuata con oltre 150 giorni di preavviso.

No of special requests:

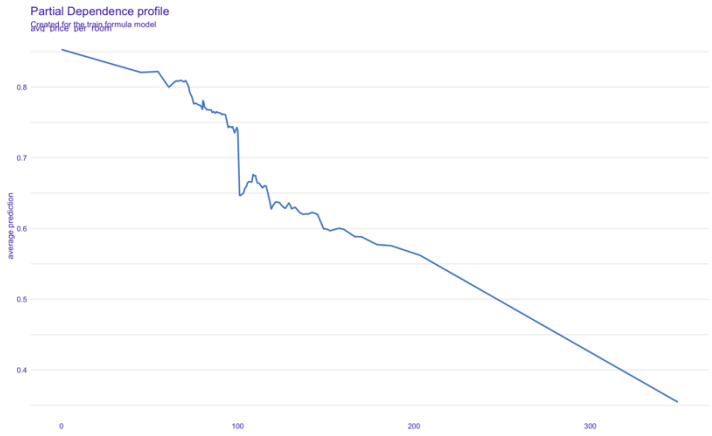
Essendo la soglia 0.35, la previsione media sembra suggerci che ogni osservazione verrà classificata come non cancellata. In realtà, l'interpretazione è che all'aumentare del numero di richieste speciali è più improbabile che la prenotazione venga classificata come "canceled".



Lorenzo Megna 868929
Massimo Trippetta 869286
Davide Vettore 868855

Averange price per room:

Allo stesso modo possiamo interpretare la dipendenza parziale con il prezzo medio per stanza come propensione del modello a classificare la prenotazione come cancellata all'aumentare del prezzo.



Applicazione del modello sui dati di scoring:

Dopo aver applicato il modello Bagging al dataset di score, osserviamo la nuova distribuzione del target :

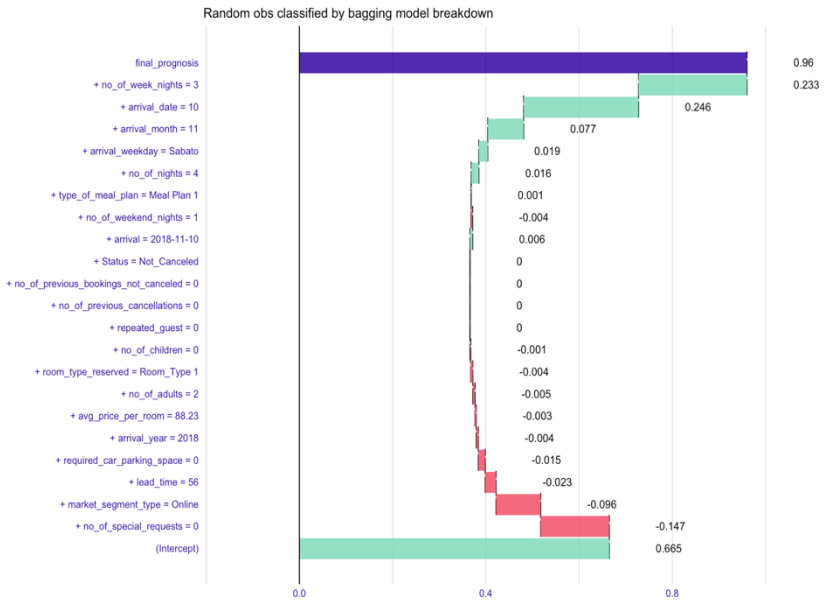
```
> prop.table(table(score$Status))

    Canceled Not_Canceled
   0.2941176    0.7058824
```

Osserviamo due prenotazioni estratte casualmente dal dataset di score e le motivazioni dietro alla classificazione del target.

Obs 1:

Questa prenotazione è classificata dal modello come non cancellata. Il valore di score è molto elevato, quindi ci aspettiamo che la previsione sia corretta. Le variabili che influiscono maggiormente sulla prognosi sono:

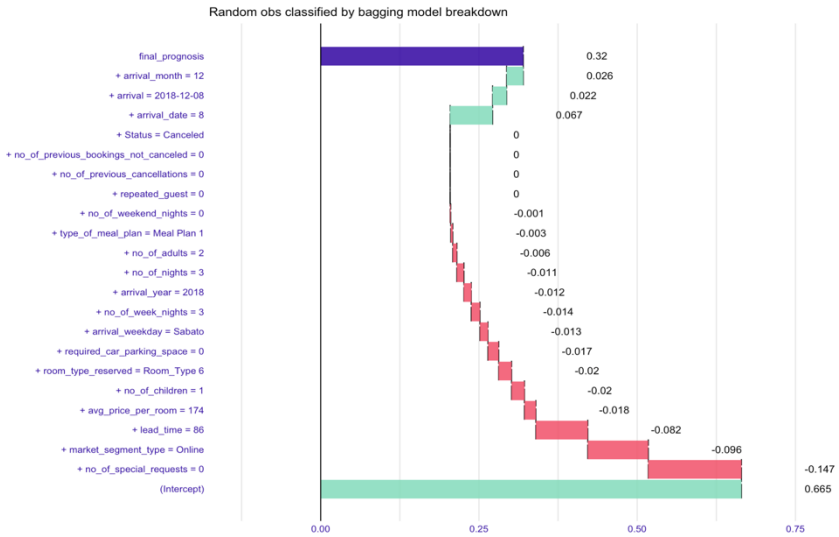
- Assenza di richieste speciali
- Prenotazione effettuata online
- Poco tempo di preavviso (56g)
+ Arrivo nel mese di novembre
+ Arrivo l'11° giorno del mese
+ Tre notti di soggiorno



Obs 2:

La seconda prenotazione estratta viene invece classificata come cancellata. Nel dettaglio:

- Il fatto che non ci siano richieste speciali contribuisce negativamente sull'esito della prenotazione.
- La prenotazione è stata effettuata online; dagli istogrammi riportati nell'introduzione sappiamo che "online" è il market segment type con la più alta frequenza di cancellazioni.
- Il tempo di preavviso è di 86 giorni; seppur non troppo elevato, è comunque sopra la media di train.



A contribuire positivamente alla prognosi finale vediamo esclusivamente la data di arrivo.
Notiamo inoltre che il valore finale predetto è di 0.32, molto vicino alla soglia. Questo significa che, se avessimo scelto una soglia leggermente inferiore, la previsione sarebbe cambiata.