14/09/2022

Esame Statistica – Master Data Science and Big Data UNIPA

N.B I codici completi sono nel file esame1.r

“A causa della concorrenza aggressiva nel settore aereo, le compagnie aeree devono concentrarsi sulla soddisfazione dei passeggeri. Il file airline permette di studiare come le compagnie aeree possono mirare a determinare l'importanza di ciascuna caratteristica nella determinazione della soddisfazione (satisfaction) dei passeggeri.”

La traccia sopracitata invita lo studente a eseguire una analisi statistica nel linguaggio R per capire quali variabili giocano un ruolo importante nell'influenzare il feedback di un passeggero verso la "soddisfazione" e prevedere se un passeggero sarà soddisfatto o meno. La variabile risposta (o dipendente) è stata individuata in “satisfaction”, mentre le variabili esplicative sono state individuate tramite regressione logistica ed albero decisionale.

Importo il dataset “airline.csv”

airline <- read.csv("airline.csv")

Divido il dataset in “Train” per il 75% e “Test” per il 25%

*nrow(airline) * 0.75*

*sample_rows <- sample(nrow(airline), nrow(airline) * 0.75)*

airline_train <- airline[sample_rows,]

airline_test <- airline[-sample_rows,]

write.csv(airline_train, "Train.csv")

write.csv(airline_test, "Test.csv")

Train = read.csv("Train.csv")

Test = read.csv("Test.csv")

Do un controllo ai dati con una summary

summary(Train)

summary(Test)

Controllo la variabile dipendente “satisfaction”

table(Train$satisfaction)

neutral or dissatisfied	*satisfied*
44058	*33870*

Converto le colonne di tipo ordinale e la variabile satisfaction in fattori. Trasformo le variabili numeriche di tipo ordinali in fattori. In R, i fattori sono utilizzati per lavorare con variabili categoriche, ovvero variabili che hanno un insieme fisso e noto di valori possibili. Sono anche utili quando si desidera visualizzare vettori di caratteri in un ordine non alfabetico. Storicamente, i fattori erano molto più facili da usare rispetto ai

caratteri. Converto anche la variabile dipendente “satisfaction” dal momento che questo mi permette di lavorarci meglio.

library(tidyverse)

Train$Inflight_wifi_service = as.factor(Train$Inflight_wifi_service),… etc

Quindi controllo l’ avvenuta trasformazione

str(Train)

Procedo ugualmente per il dataset Test

Test$Inflight_wifi_service = as.factor(Test$Inflight_wifi_service),….etc

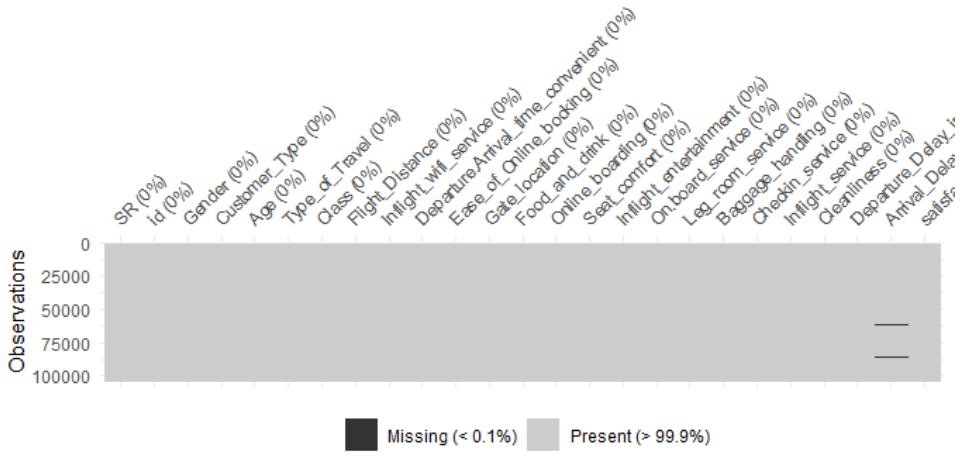
str(Test)

Controllo se vi sono valori nulli nel dataset originale airline tramite la libreria visdat e la funzione vis_miss

install.packages("visdat")

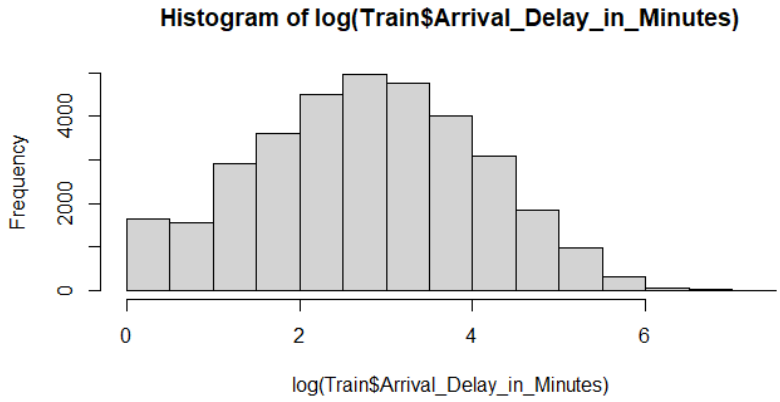
library(visdat)

vis_miss(airline, warn_large_data = FALSE)

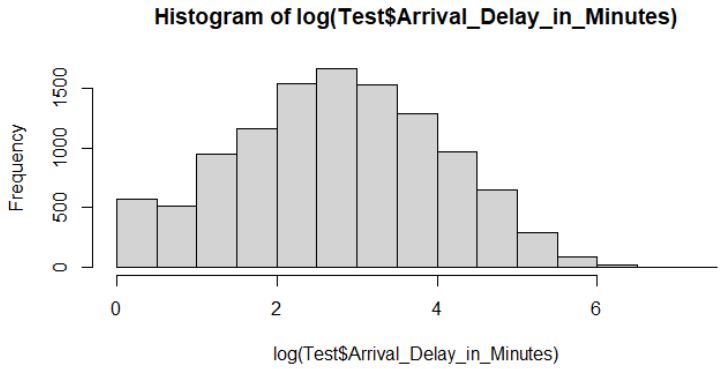


Noto che ci sono NA nella colonna Arrival_Delay _in_Minutes quindi decido di vedere la distribuzione di NA in train e test tramite un istogramma

hist(log(Train$Arrival_Delay_in_Minutes))



hist(log(Test$Arrival_Delay_in_Minutes))



Sostituisco la colonna con valori NA creando prima la copia dei due dataset e poi sostituendo gli NA con la media della colonna

Train.avg = Train

summary(Train.avg)

Test.Avg = Test

summary(Test.Avg)

NA.position <- which(is.na(Train.avg$Arrival_Delay_in_Minutes))

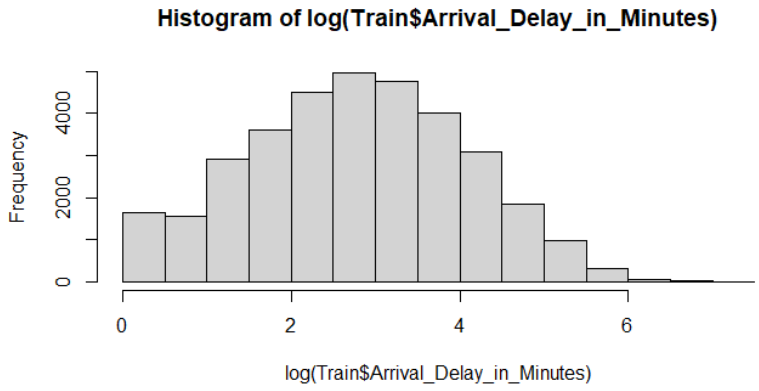
Train.avg$Arrival_Delay_in_Minutes[NA.position] = mean(Train.avg$Arrival_Delay_in_Minutes, na.rm = TRUE)

NA.position1 <- which(is.na(Test.Avg$Arrival_Delay_in_Minutes))

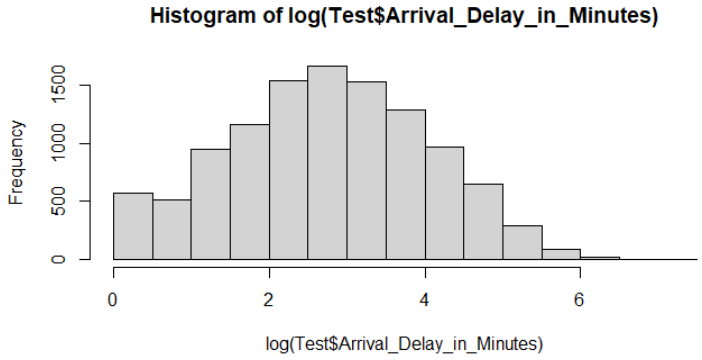
Test.Avg$Arrival_Delay_in_Minutes[NA.position1] = mean(Test.Avg$Arrival_Delay_in_Minutes, na.rm = TRUE)

Controllo che il grafico abbia la stessa distribuzione dopo la sostituzione

hist(log(Train$Arrival_Delay_in_Minutes))



hist(log(Test$Arrival_Delay_in_Minutes))



Verifichiamo non ci siano più NA

> sum(is.na(Train.avg))

[1] 0

> sum(is.na(Test.Avg))

[1] 0

Modello di regressione logistica, carichiamo le librerie necessarie.

La regressione consiste nel prevedere un risultato y di una variabile risposta, utilizzando uno o più predittori utilizzando delle variabili esplicative. In R, la regressione logistica utilizza la funzione glm in cui si specifica la variabile dipendente prima della tilde e successivamente le variabili esplicative. Il parametro "family" specifica il tipo di modello. In questo caso "binomial" indica ad R di eseguire una regressione logistica. Una volta costruito il modello, è possibile utilizzarlo per stimare le probabilità. Fornendo il parametro type = "response" alla funzione predict, si ottengono le probabilità previste, più facili da interpretare. Quindi carichiamo le librerie necessarie e proseguiamo.

install.packages('broom')

install.packages('tidyverse')

install.packages('caret')

install.packages('MASS')

install.packages('ROCR')

library(broom)

library(tidyverse)

library(caret)

library(MASS)

library(ROCR)

Imposto il seed per la randomizzazione.

Creo il modello di regressione logistica con tutte le variabili.

Da un errore presentatomi con la funzione predict (successiva), era indicato che la colonna Gate_location avesse livelli diversi in Train e Test. Quindi, prima della funzione predict, procedo a unire i valori delle colonne Gate_location

mylogit$xlevels[["Gate_location"]] <- union(mylogit$xlevels[["Gate_location"]], levels(Test.Avg$Gate_location))

Controllo il summary del modello.

Ricreo il modello con le variabili più importanti, eliminando Flight_distance poiché irrilevante.

Faccio la summary.

Faccio la predizione con il Test dataset.

predict = predict(mylogit, type = 'response' , newdata=Test.Avg)

summary(predict)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.000000 0.002955 0.454300 0.493319 0.999998 1.000000

Controllo le probabilità di predizione con una summary

summary(predict)

Controllo la probabilità dell' output usando la media

tapply(predict, Test.Avg$satisfaction, mean)

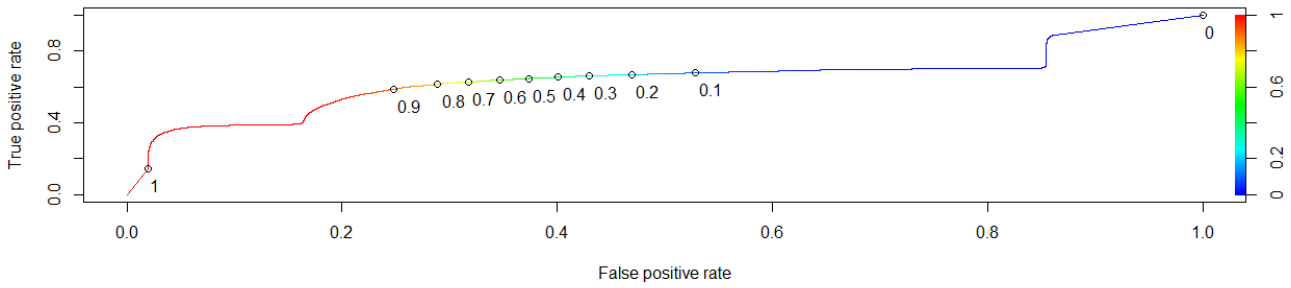
neutral or dissatisfied satisfied

0.3836703 0.6390028

Costruisco la curva ROC

La curva ROC permette di comprendere meglio la capacità di un modello di distinguere tra previsioni positive e negative dell' esito di interesse rispetto a tutti gli altri. La curva ROC illustra la relazione tra la percentuale di esempi positivi e negativi. Più la curva ROC è sovrapposta su una diagonale che divide in due il grafico, più il modello è poco prestante perché non riesce a dividere tra gradi positivi e negativi. Più la curva si allontana dalla diagonale immaginaria centrale, più sono migliori le prestazioni del modello. Per quantificare la prestazione, viene usato l' AUC (Area Under the Curve). Il modello con basse prestazioni ha AUC = 0.50 perché corrisponde alla diagonale, mentre un modello perfetto ha AUC = 1.00. Nel mio caso,

come nella maggior parte dei casi, l' AUC si trova a metà strada e più è vicino al valore 1.00, meglio è, come in questo caso.



Calcolo l' accuratezza con la matrice di confusione. Dalla curva ROC notiamo come la soglia ottimale per il cut sia 0.7. L' output della matrice indica quattro possibili risultati ossia:

	Valore osservato falso	Valore osservato vero
Predizione falsa	**corretto**	**Falso negativo**
Predizione vera	**Falso positivo**	**corretto**

Nel nostro caso ci sono 10116 corretti neutrali o insoddisfatti, 4705 falsi neutrali o insoddisfatti, 4150 falsi soddisfatti, 7005 veri soddisfatti:

table(Test.Avg$satisfaction, predict > 0.7)

FALSE TRUE

neutral or dissatisfied 10116 4705

satisfied 4150 7005

Calcolo l' accuratezza

L' accuratezza è calcolata con la somma tra veri neutrali o insoddisfatti più veri soddisfatti, diviso il numero totale di osservazioni. Più alta è, meglio è.

Accuracy_avg_Logit = (10116+7005)/(10116+4705+4150+7005)

Accuracy_avg_Logit

0.6591084

Calcolo la sensibilità

La sensibilità riguarda la percentuale di osservazioni in cui la risposta effettiva era vera e in cui il modello aveva previsto che fosse vera. Si calcola ponendo i veri soddisfatti diviso la somma tra falsi soddisfattui e veri soddisfatti. Più è alto il valore meglio è. In questo caso 7005 su 11.155 clienti soddisfatti sono stati correttamente previsti.

Recall = (7005)/(7005+4150)

Recall

0.6279695

Calcolo la specificità

Si tratta della proporzione di osservazioni in cui la risposta effettiva è risultata di veri neutrali o insoddisfatti e in cui il modello ha previsto la quantità di neutrali o insoddisfatti.

Specificity = (10116)/(10116+4705)

Specificity

0.682545

Calcolo il valore AUC

L' AUC qui ottenuto ci dice che il modello è prestante perché sopra 0.5

AUC = as.numeric(performance(ROCRpred, "auc")@y.values)

AUC

0.645178

Costruisco il modello dell' albero decisionale

L' albero decisionale permette di modellare la relazione tra predittori e risultato di interesse. Utilizzo il pacchetto rpart e le funzioni relative. Con la funzione varImp individuo le variabili più significative, attraverso cui ricalcolare un nuovo modello e creare l' albero decisionale finale.

install.packages("rpart")

library(rpart)

library(rpart.plot)

tree = …

Analizzo l' importanza delle variabili usando la funzione varImp

varImp(tree)

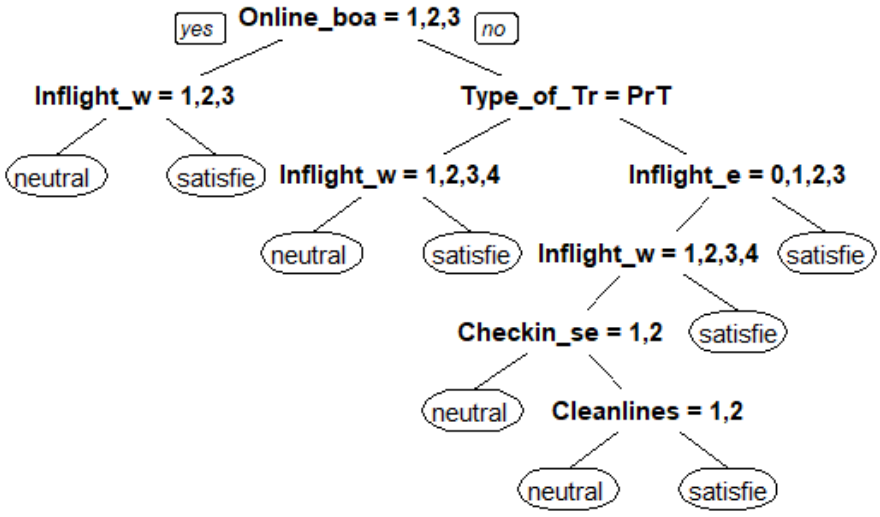
Ricalcolo il modello con le variabili più significative

tree1 = ..

summary(tree1)

Visualizzo l' albero decisionale

prp(tree1)



Ottimizzo iniziando con la cross-validation

Definisco

numFolds = trainControl(method = "cv", number = 10)

cpGrid = expand.grid(.cp = seq(0.01,0.5,0.01))

Avvio la cross-validation

Adesso stimo la bontà dell' albero attraverso la stima per cross-validation attraverso cui comprendo il livello ottimale di complessità dell' albero. Successivamente creo un nuovo modello e predico i valori.

train…

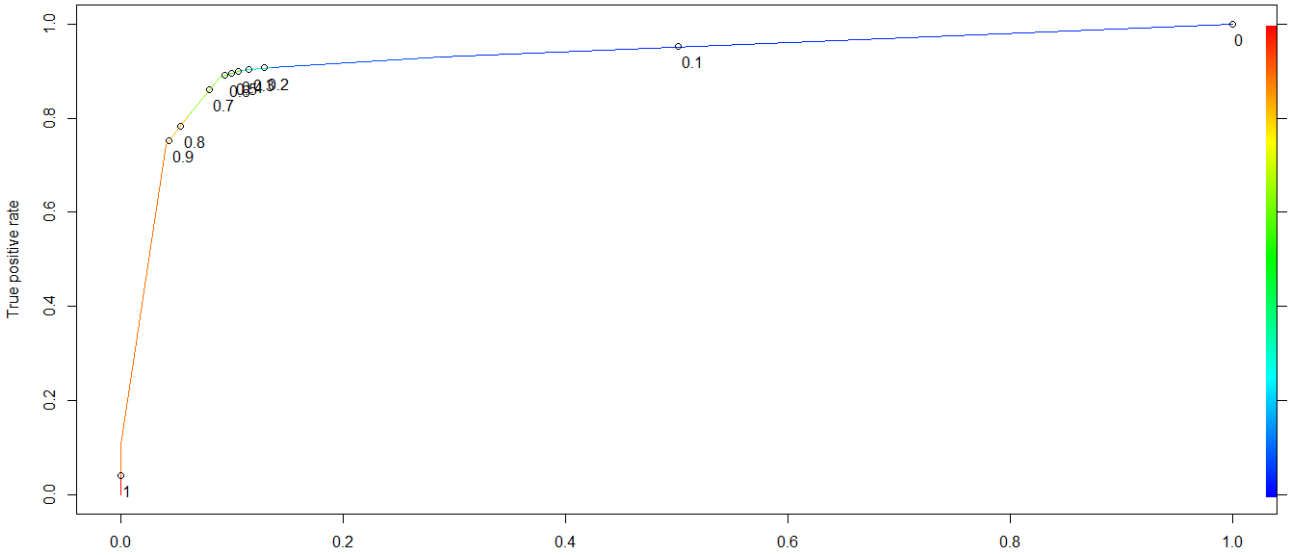
Creo un nuovo modello CART

tree1 =

Predico i valori sul dataset Test

predictROC = predict(tree1, newdata = Test.Avg)

Calcolo la curva ROC



Anche qui vediamo che 0.7 è la migliore soglia per tagliare e calcoliamo la matrice di confusione

table(Test.Avg$satisfaction, PredictROC[,2] > 0.7)

	FALSE	*TRUE*
neutral or dissatisfied	*13920*	*901*
satisfied	*2144*	*9011*

Calcolo l' accuratezza

Accuracy_avg_Tree = (13920+9011)/(13920+9011+901+2144)

Accuracy_avg_Tree

0.8827764

Calcolo la sensibilità

Recall = (9011)/(9011+2144)

Recall

0.8077992

Calcolo la specificità

Specificity = (13920)/(13920+901)

Specificity

0.9392079

Calcoliamo il valore AUC

AUC = as.numeric(performance(pred, "auc")@y.values)

AUC

0.9265709

Guardando ai risultati di entrambi i modelli, posso concludere che l' albero decisionale ha avuto risultati migliori rispetto alla regressione logistica. Guardando ai valori di accuratezza e specificità, l' albero decisionale ha ottenuto punteggi superiori. Si conclude che la soddisfazione dei clienti dipende da diversi fattori da me individuati e analizzati.

Pietro Enea