VPC 2.0

Manuale installazione

e utilizzo

[1 Introduzione 2](#_Toc108711031)

[2 Requisiti ed installazione di R e delle estensioni di R 3](#_Toc108711032)

[2.1 Requisiti 3](#_Toc108711033)

[2.2 Installazione R 3](#_Toc108711034)

[2.3 Installazione estensioni di R e software VPC 2.0 3](#_Toc108711035)

[3 Utilizzo del software 5](#_Toc108711036)

[3.1 Modello attuale 5](#_Toc108711037)

[3.2 Retrain del modello 5](#_Toc108711038)

# Introduzione

Bain ha sviluppato un tool….

# Requisiti ed installazione di R e delle estensioni di R

## Requisiti

Windows 10 Enterprise 64 bit

## Installazione R

Scaricare la versione di R che si desidera installare dalla pagina https://cran.r-project.org/bin/windows/base/

In accordo con i test effettuati, suggeriamo di scaricare la versione 4.1.1, reperibile all’indirizzo <https://cran.r-project.org/bin/windows/base/old/4.1.1/>, file R-4.1.1-win.exe.

Eseguire il file .exe che farà partire l’installazione di R.

Per poter usufruire di una agevole interfaccia grafica per l’esecuzione dei comandi R, scaricare e installare anche il software RStudio:

Sempre in accordo con i test effettuati, suggeriamo la versione 1.4 reperibile all’indirizzo <https://dailies.rstudio.com/version/1.4.1743-4/>, file RStudio-1.4.1743.exe relativo alla “Platform” Windows 10+, sebbene anche versioni più recenti di RStudio dovrebbero funzionare senza problemi.

## Installazione estensioni di R e software VPC 2.0

Una volta terminata l’installazione:

* aprire una sessione di RStudio;
* posizionarsi nella “console” di RStudio, come mostrato dalla freccia in Figura 1, e digitare il comando  
  getwd()

Questo permetterà di vedere qual è la directory di lavoro di RStudio (ad es. “C:/nomeutente/Documents”).

* scaricare il file .zip contenente il software VPC\_2.0 ed estrarlo nella stessa directory output del comando precedente (o, eventualmente in una sotto directory creata appositamente).

Graphical user interface, application

Description automatically generated

Figura : Interfaccia RStudio

A questo punto bisogna installare le estensioni di R. Questo può essere fatto in due modi (notare che la macchina deve essere connessa a internet).

Facendo girare lo script già pronto:

* cambiare la working directory di RStudio con il comando

setwd(“C:/nomeutente/Documents/codice\_produzione/codici”)

dove, al posto di “C:/nomeutente/Documents” si dovrà inserire la working directory vista precedentemente, più eventualmente le sottocartelle create. Per arrivare alla cartella estratta dallo .zip si può utilizzare il comando Tab posizionadosi prima delle virgolette di chiusura e scegliendo ricorsivamente tra le directory proposte fino ad arrivare a quella contenente “codici”.

* Lanciare nella console il comando  
  source(“install\_packages.R”)

Dando i comandi da riga di comando:

* Dalla tab File aprire il file “install\_packages.R”.
* Fare copia incolla del testo del file nella console RStudio e dare invio.

# Utilizzo del software

Il software ha tre funzionalità principali:

* caricare il modello di albero decisionale già addestrato e modificato secondo le logiche definite con l’utente (ovvero quello che verrà poi messo in produzione);
* effettuare il predict su nuovi dati di test utilizzando il modello già addestrato;
* addestrare un albero decisionale basato su nuovi dati di input.

Prima di eseguire qualsiasi script è sempre bene rimuovere dalla sessione R eventuali oggetti precedenti, tramite il comando “rm(list = ls())” o utilizzando il tasto apposito presente nell’interfaccia

Graphical user interface, text, application, email

Description automatically generated

## Modello attuale

1. I codici necessari per caricare i modelli già addestrati, sia SME sia Corporate, si trovano rispettivamente nei file:

* SME\_run\_model\_asis.R
* CO\_run\_model\_asis.R

1. Per farli girare occorre utilizzare il comando source, come visto nel capitolo precedente; ad esempio, per il corporate:
2. source(“CO\_run\_model\_asis.R”)
3. Alla fine del run, viene prodotta nel path “output/asis/co” una cartella il cui nome sarà dato dalla data in cui viene fatto girare il codice (nel formato YYYYMMGG) e al cui interno si trovano i vari output, ovvero:

* plot della struttura dell’albero decisionale;
* plot della curva ROC;
* plot della feature importance;
* file tree\_co\_asis.xml contenente l’albero decisionale in formato testuale, necessario per la traduzione tramite Drools;
* file results\_co.xlsx con i cluster giallo, rosso e verde definiti in base alle soglie inserite
* file performance\_co.xlsx contenente una tabella che mostra per singola foglia lo score e le numeriche del campione all’interno di quella foglia, utile per definire le soglie di taglio;
* file train\_co\_asis.csv contenente il db di train
* file test\_co\_asis.csv contenente il db di test e la colonna con lo score predetto dall’albero.

1. Il codice legge il file in formato *csv* “master\_asis.csv”, contenuto nella cartella “data/master/co”, che non deve mai essere modificato o sostituito. Tale file contiene lo storico analizzato e processato fino al 30 giugno 2022 secondo le logiche condivise.

## Predict

1. All’interno della cartella “codici” sono presenti due file per il predict su sme e corporate. Prima di procedere con l’esecuzione dello script tramite il comando “source(‘SME\_predict.R’)” da lanciare in R, è necessario definire il nome del file contenente i dati da testare che deve essere salvato nella cartella “data/test/sme” in formato *csv*. Il codice da eseguire è quindi:
2. nome\_db\_test <- 'dt\_test\_sme.csv'
3. source(‘SME\_predict.R’)
4. Con l’esecuzione di questi comandi, all’interno della cartella ‘output/test/sme/YYYYMMGG’ vengono creati i seguenti file:

* 'dt\_test\_sme\_pred.csv’ contenente una copia dei dati in input ai quali viene aggiunta la colonna con lo score restituito dall’albero;
* ‘results\_sme\_test.csv’ contenente le numeriche di gialli, rossi e verdi in base alle soglie impostate nel file di configurazione;
* ‘sme\_ROC\_test.pdf’ contenente la curva roc associata al nuovo dataset di test.

1. Nella fase di test è fondamentale che la tabella in input contenga tutte le variabili utilizzate nel training con la stessa nomenclatura, rispettando maiuscole e minuscole e con gli stessi formati. Questi dettagli sono contenuti nell’excel allegato “calcolo\_variabili.xlsx” insieme alle indicazioni su come riempire i missing (colonna “fill missing test”). Per un ulteriore analisi si può provare a testare il modello riempiendo solo i missing della colonna “fill missing DSI”, ovvero riproducendo la situazione che si avrà in produzione. Chiaramente se il db di test ha le stesse percentuali e logiche di missing del db utilizzato per il train, il primo approccio è quello che darà le performance reali; nel caso in cui i dati provenissero da una storicizzazione futura in cui il numero dei missing risulta notevolmente ridotto, il secondo approccio è quello consigliato.

## Retrain del modello

1. Il codice del retrain serve a generare un nuovo albero (sme o corporate) sulla base di nuovi dati. Prima di essere inserito come input al modello, è necessaria un’analisi dettagliata del nuovo dataset, per verificare che le informazioni siano consistenti ma soprattutto per capire come gestire al meglio i missing presenti.
2. Ad esempio possono esserci casi in cui il missing rappresenta un’informazione (sconfino =0, cliente non censito, ecc..) per cui ha senso riempire il valore mancante con tale informazione. In altri casi invece in cui non è chiara l’origine del missing, è preferibile non intervenire per evitare di introdurre bias nel dataset.

### Dati di input

I nuovi dati per il riaddestramento devono essere salvati in un file nominato “master\_retrain.csv” nella cartella “data/master/segmento”. Affinché il modello funzioni, tale tabella deve contenere i campi “NUM\_PRATICA” e “PROGRESS\_PRATICA” come chiave del db e il campo “target” per la valutazione della pratica. Per quanto riguarda tutte le altre variabili di merito, è importante che esse vengano inserite nel file di configurazione (config/ co\_config\_file\_retrain.R) con la medesima nomenclatura, rispettando maiuscole e minuscole.

Esempio della tabella:

|  |  |  |
| --- | --- | --- |
| 1. CAMPO | 1. DATATYPE | 1. DESCRIZIONE |
| 1. NUM\_PRATICA | 1. Numero | 1. Identificativo della pratica |
| 1. PROGRESS\_PRATICA | 1. Numero | 1. Identificativo della pratica |
| 1. target | 1. Stringa | 1. “OK” o “DEFAULT” |
| 1. Var\_1 | 1. Numero | 1. Numero di fidi |
| 1. Var\_2 | 1. Stringa | 1. “TRUE” o “FALSE” |

Esempio del corrispondente file di configurazione:

cols\_modello <- c(

'NUM\_PRATICA',

'PROGRESS\_PRATICA',

'target',

‘Var\_1’,

‘Var\_2’)

1. Una volta verificate queste condizioni, si può eseguire lo script tramite il comando
2. source(“CO\_run\_model\_retrain.R”)
3. All’interno del file di configurazione sono presenti alcuni parametri che permettono di attivare/disattivare sia il processo di pruning manuale che l’ottimizzazione automatica.   
   Consigliamo di lasciare i flag impostati come segue:
4. flag\_prune <- FALSE
5. flag\_opt <-TRUE
6. Alla fine del run, viene prodotta nel path “output/retrain/co” una cartella il cui nome sarà dato dalla data in cui viene fatto girare il codice (nel formato YYYYMMGG) e al cui interno si trovano i vari output, ovvero:

* plot della struttura dell’albero decisionale;
* plot della curva ROC;
* plot della feature importance;
* file tree\_co.xml contenente l’albero decisionale in formato testuale, necessario per la traduzione tramite Drools;
* file results\_co.xlsx con i cluster giallo, rosso e verde definiti in base alle soglie inserite
* file performance\_co.xlsx contenente una tabella che mostra per singola foglia lo score e le numeriche del campione all’interno di quella foglia, utile per definire le soglie di taglio;
* file train\_co.csv contenente il db di train;
* file test\_co.csv contenente il db di test e la colonna con lo score predetto dall’albero.

1. Il modello prodotto viene salvato nella cartella “codici/modelli” con il nome “modello\_co.RDS”.

## File di configurazione

1. Tutti i file di configurazione sono contenuti nella cartella “config”. Al loro interno vengono definiti:

* Le variabili del modello;
* Le soglie di taglio per i colori;
* Gli iperparametri del modello;
* Alcuni path di riferimento;
* Flag di ottimizzazione (solo per retrain);
* Flag per il pruning manuale (solo per retrain.

1. Per quanto riguarda il modello asis tali file non vanno modificati.
2. Per il retrain invece è necessario allineare la lista delle variabili con quelle inserite nel db per l’addestramento e gli altri parametri possono essere cambiati in base alle necessità.

## Altri file e cartelle

1. Oltre ai file già citati, all’interno della cartella “codici” sono presenti:

* la cartella “modelli” dove vengono letti e salvati i modelli prodotti dagli script;
* il file “pruning.R” che contiene il codice per il pruning manuale di un albero;
* il file “model\_function.R” che contiene le funzioni necessarie richiamate dagli script.