Msc.: Applied Data Science for Banking and Finance

Course: LABORATORY OF DATA ANALYTICS FOR BANKING AND INSURANCE

Dataset: *Banking Marketing Target*

Gislon Chiara **ID student: 5109831**

Degli Agosti Aron **ID student: 5114195**

Introduction

The data is related to the direct marketing campaigns (phone calls) for term deposits of a Portuguese banking institution.

Term deposits are a major source of income for a bank. A term deposit is a cash investment held at a financial institution. The money is invested for an agreed rate of interest over a fixed amount of time, or term. The bank has various outreach plans to sell term deposits to their customers such as email marketing, advertisements, telephonic marketing, and digital marketing.

Telephonic marketing campaigns still remain one of the most effective way to reach out to people. However, they require huge investment as large call centers are hired to actually execute these campaigns. Hence, it is crucial to identify the customers most likely to convert beforehand so that they can be specifically targeted via call.

The classification goal is to predict if the client will subscribe to a term deposit.

The attributes:

1. age (numeric)
2. job : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student", "blue-collar","self-employed","retired","technician","services")
3. marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)
4. education (categorical: "unknown","secondary","primary","tertiary")
5. default: has credit in default? (binary: "yes","no")
6. balance: average yearly balance, in euros (numeric)
7. housing: has a housing loan? (binary: "yes","no")
8. loan: has a personal loan? (binary: "yes","no")

Related with the last contact of the current campaign:

9. contact: contact communication type (categorical: "unknown","telephone","cellular")
10. day: last contact day of the month (numeric)
11. month: last contact month of year (categorical: "jan", "feb", "mar", …, "nov", "dec")
12. duration: last contact duration, in seconds (numeric)

Other attributes:

13. campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
14. pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
15. previous: number of contacts performed before this campaign and for this client (numeric)
16. poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")

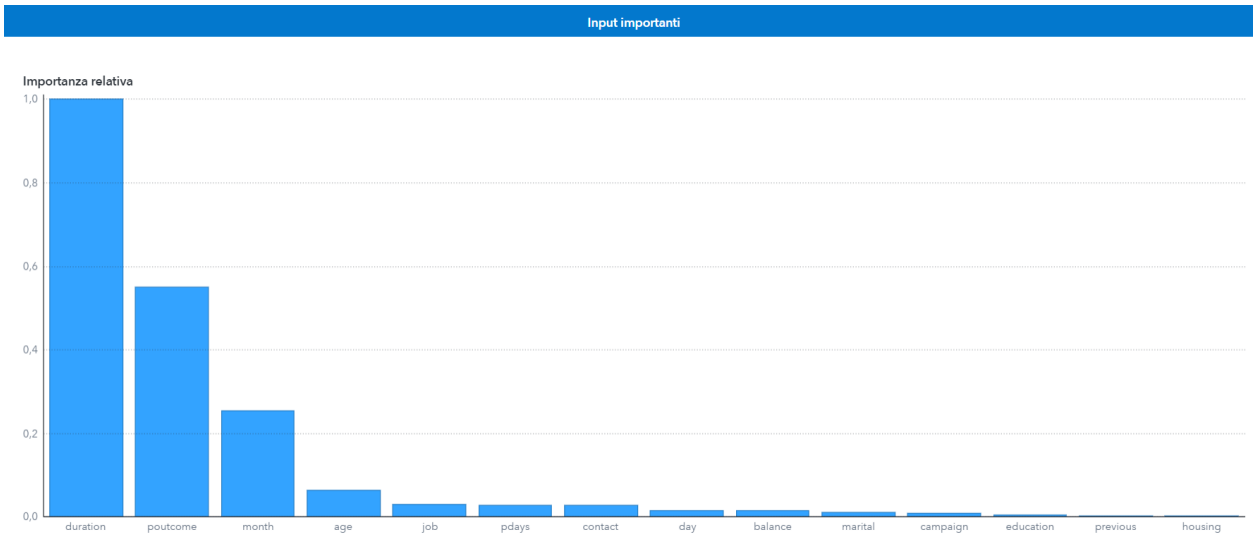
Output variable (target):

17. y: has the client subscribed a term deposit? (binary: "yes","no")

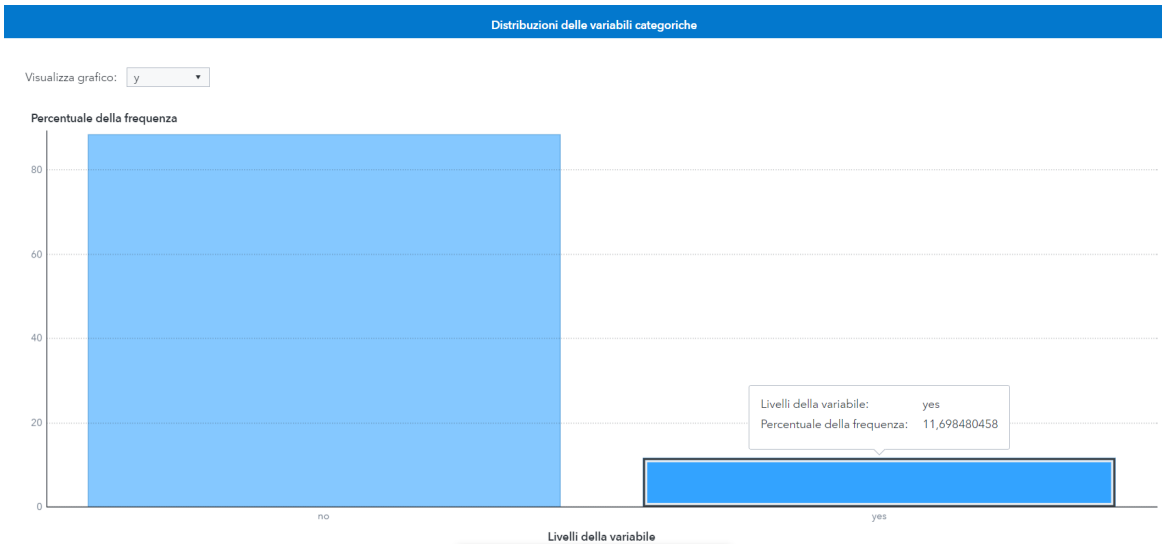
Data Exploration

We initially split the dataset into two partitions: 70% of data for training and 30% for validating the model.

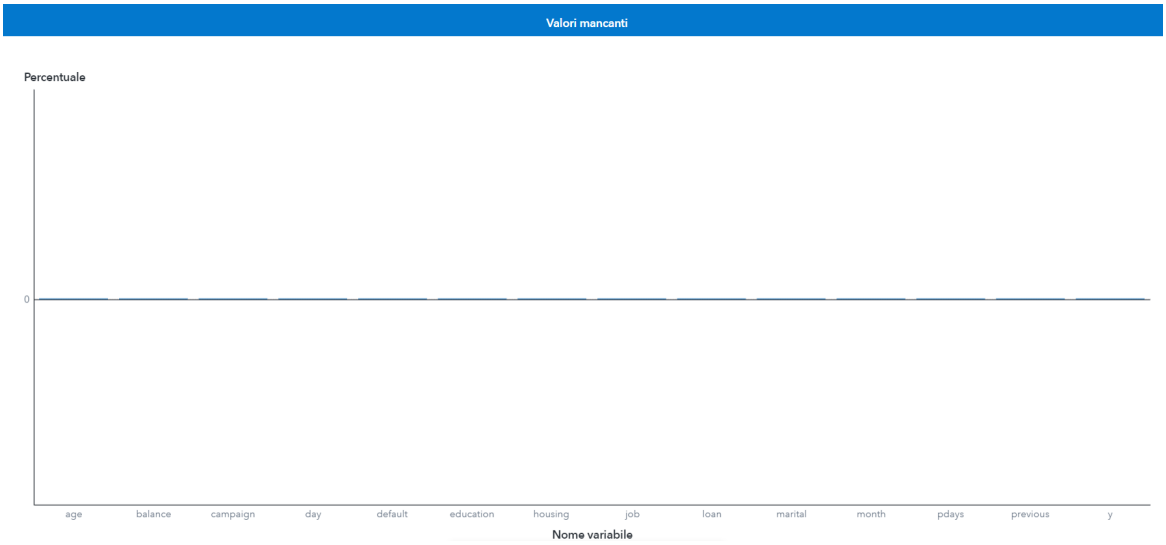
We started by adding a Data Exploration Node to analyze our dataset. The chart below shows us the most important input variables - it is determined through the decision tree method. It is clear that the most important variables are ‘duration’, ‘poutcome’ and ‘month’.



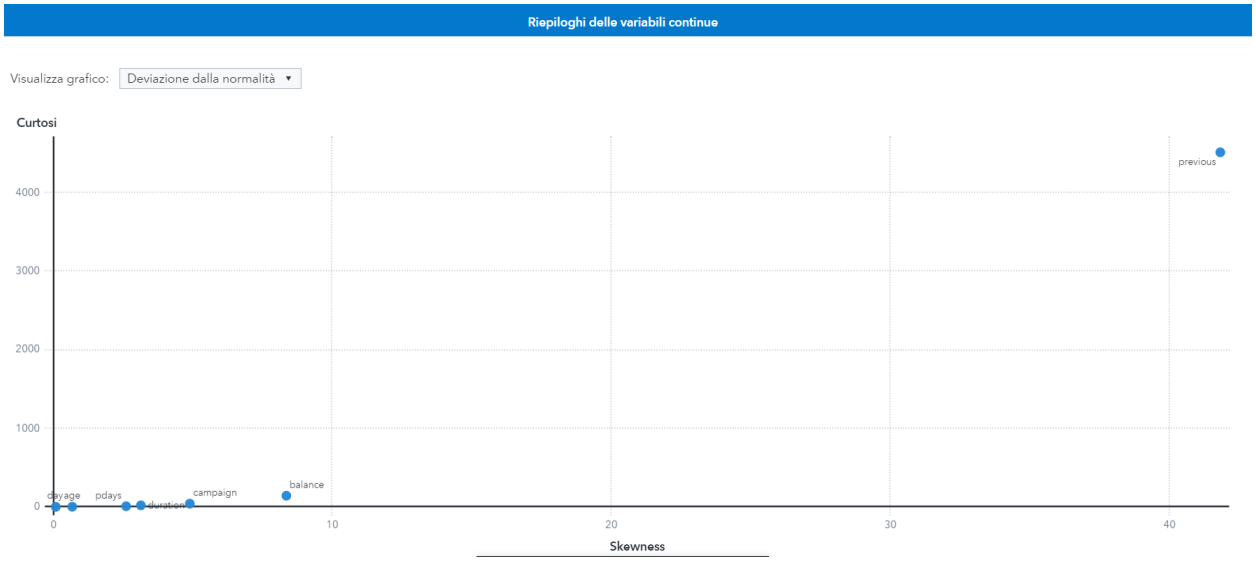
The bar plot below shows the distribution of our target variable “y”, which indicates if the client has subscribed to the term deposit. In 11,7% of the total cases the client has responded positively to the marketing campaign.



Fortunately, we don’t have any missing values, so we don’t need to replace any values. In the opposite case, we would have to add a replacement node to solve this problem.



Now we can move on to analyze the skewness and kurtosis chart. As we can see there are different variables with high skewness and/or really high kurtosis, specifically, the ‘previous’ variable.



Momenti delle variabili continue

Nome variabile	Minimo	Massimo	Media	Deviazione st...	Skewness	Curtosi	Variabilità rela...	Media più 2 SD	Media meno 2...
age	18	95	40,9362	10,6188	0,6848	0,3196	0,2594	62,1737	19,6987
balance	-8.019	102.127	1.362,2721	3.044,7658	8,3603	140,7515	2,2351	7.451,8037	-4.727,2596
campaign	1	63	2,7638	3,0980	4,8987	39,2497	1,1209	8,9599	-3,4322
day	1	31	15,8064	8,3225	0,0931	-1,0599	0,5265	32,4514	-0,8385
duration	0	4.918	258,1631	257,5278	3,1443	18,1539	0,9975	773,2187	-256,8925
pdays	-1	871	40,1978	100,1287	2,6157	6,9352	2,4909	240,4553	-160,0597
previous	0	275	0,5803	2,3034	41,8465	4.506,8607	3,9692	5,1872	-4,0266

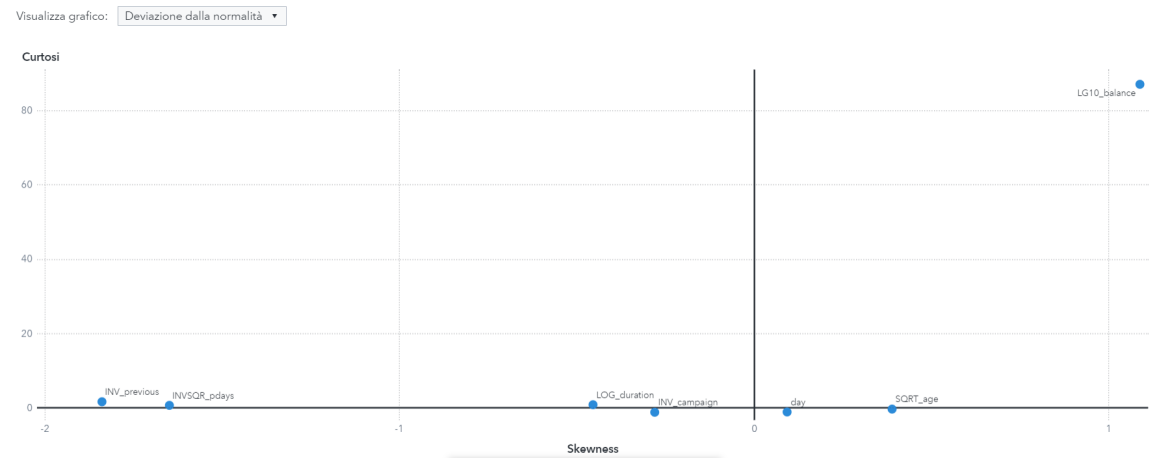
Since the skewness and the kurtosis are so high, we decided to rescale some variables using the "*best*" function through the transformation node: it selected the best transformation function for each variable.

Momenti delle variabili continue

Nome variabile	Minimo	Massimo	Media	Deviazione st...	Skewness	Curtosi	Variabilità rela...	Media più 2 SD	Media meno 2...
INVSQR_pdays	0,0000	0,5000	0,4091	0,1927	-1,6484	0,7195	0,4710	0,7944	0,0237
INV_campaign	0,0156	0,5000	0,3492	0,1374	-0,2801	-1,1368	0,3935	0,6239	0,0744
INV_previous	0,0036	1	0,8772	0,2667	-1,8385	1,6856	0,3040	1,4106	0,3438
LG10_balance	0	5,0420	3,9602	0,0925	1,0878	87,0517	0,0234	4,1452	3,7752
LOG_duration	0	8,5009	5,1718	0,9218	-0,4542	0,8884	0,1782	7,0155	3,3282
SQRT_age	4,3589	9,7980	6,4257	0,8043	0,3890	-0,2859	0,1252	8,0343	4,8171
day	1	31	15,8064	8,3225	0,0931	-1,0599	0,5265	32,4514	-0,8385

Now all the variables considered for the transformation have a better skewness and kurtosis.

Riepiloghi delle variabili continue



The Imputation Node rejects the old variables and substitutes them with the transformed ones. Then, we added a Variable Selection Node which rejected a few variables:

DEFAULT		BINARY	REJECTED	Criterio di combinazione
EDUCATION		NOMINAL	REJECTED	Criterio di combinazione
INVSQR_PDAYS	Trasformazione pdays	INTERVAL	REJECTED	Criterio di combinazione
INV_CAMPAIGN	Trasformazione campaign	INTERVAL	REJECTED	Criterio di combinazione
SQRT_AGE	Trasformazione age	INTERVAL	REJECTED	Criterio di combinazione

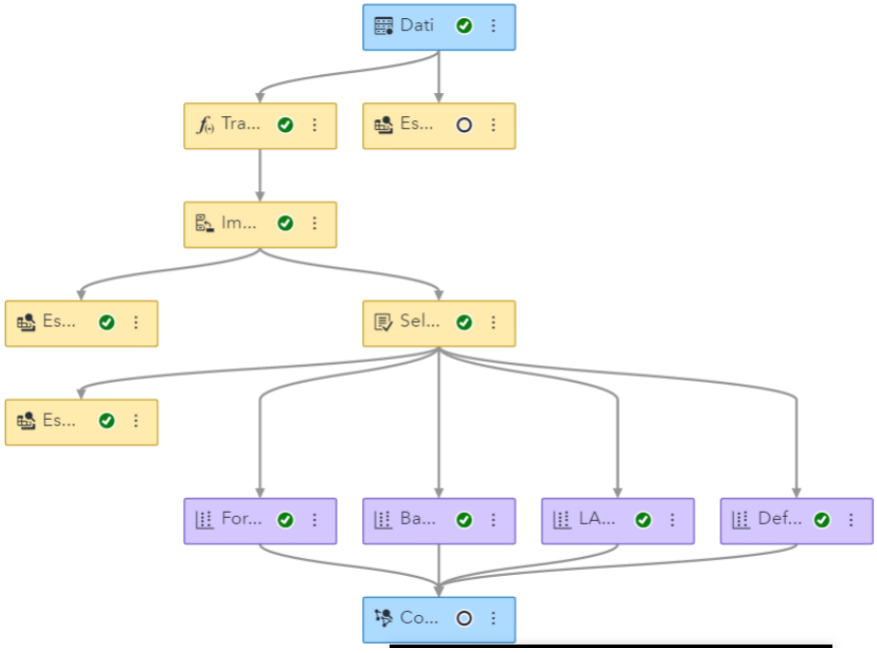
The tables below show the variable importance before and after the Variable Selection Node:

Importanza della variabile			
	Training		
Variabile	Importanza	Importanza relativa	Conteggio
LOG_duration	1567,03	1,0000	22
poutcome	879,32	0,5611	2
month	387,91	0,2475	18
SQRT_age	114,00	0,0728	12
job	58,3432	0,0372	18
INVSQR_pdays	50,2762	0,0321	6
LG10_balance	33,6153	0,0215	17
contact	26,8216	0,0171	2
day	14,9727	0,0096	9
INV_previous	13,6309	0,0087	6
INV_campaign	12,6049	0,0080	6
marital	8,5471	0,0055	5
education	5,3865	0,0034	2
loan	2,0370	0,0013	1
housing	0,9377	0,0006	2

Importanza della variabile			
	Training		
Variabile	Importanza	Importanza relativa	Conteggio
LOG_duration	1605,55	1,0000	28
poutcome	880,67	0,5485	3
month	448,15	0,2791	22
job	88,5628	0,0552	23
contact	72,3281	0,0450	6
LG10_balance	48,5559	0,0302	26
day	46,2793	0,0288	15
INV_previous	24,9020	0,0155	9
housing	18,2661	0,0114	3
loan	10,3297	0,0064	2
marital	4,8875	0,0030	4

Logistic Regression

Regression analysis is a procedure that calculates the likelihood of the outcome variable based on a linear combination of the predictors. Binary logistic regression is a regression model in which the outcome variable can only assume two values: 0 or 1.



We decided to confront the Default Logistic Regression Node with Forward, Backward and LASSO selection methods. To further improve our models, we changed the 'model selection' and the 'selection-process stopping' criteria to 'validation ASE', which takes the minimum Average Squared Error on the validation data.

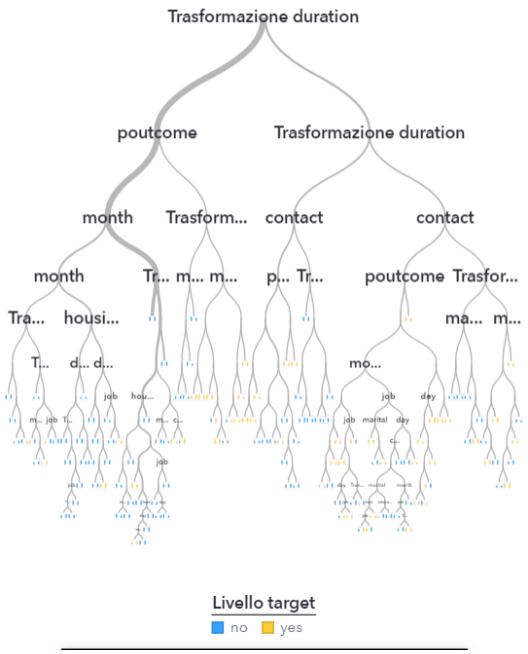
Confronto di modelli

Champi...	Nome	Nome a...	KS (You...	Accurat...	Averag...	Area sot...	Lift cum...	Percent...	Cutoff	Ruolo d...	Profond...	Score F1	False Di...
*	Forward	Regressione logistica	0,6653	0,9028	0,0700	0,9054	4,9842	49,8425	0,5000	VALIDATE	10	0,4675	0,3494
	Backward	Regressione logistica	0,6653	0,9028	0,0700	0,9054	4,9842	49,8425	0,5000	VALIDATE	10	0,4675	0,3494
	LASSO	Regressione logistica	0,6442	0,9020	0,0715	0,8985	4,8141	48,1411	0,5000	VALIDATE	10	0,4214	0,3183
	Default LR	Regressione logistica	0,6589	0,9026	0,0706	0,9033	4,9275	49,2754	0,5000	VALIDATE	10	0,4667	0,3506

The champion model is the 'Forward' logistic regression.

Tree Based Models

A decision tree is a computational approach employed in machine learning to handle both classification and regression assignments. It's called a "tree" because it's structured like a tree diagram, with branches and nodes. The main purpose of a decision tree is to break down complex problems into smaller, more manageable sub-problems.



In order to boost the performance of the model, we opt to divide the categorical variable. There are several options to choose from:

- Chi-square: employs a chi-square statistic to divide each variable, and subsequently employs the corresponding p-values derived from the splits to determine the dividing variable.
- Gini: employs the reduction in the Gini index to divide each variable and determine the division.
- Entropy: employs the information gain or decrease in entropy to divide each variable and determine the division. A minimum threshold for the decrease in entropy or increase in information gain can be specified.

As we can see from the table below, the Gini tree was the best one among all of them before pruning.

Confronto di modelli

Champi...	Nome	Nome a...	KS (You...	Accurat...	Averag...	Area sot...	Lift cum...	Percent...	Cutoff	Ruolo d...	Profond...	Score F1	False Di...
★	Gini Tree	Albero decisionale	0,6531	0,9075	0,0698	0,8953	5,1335	51,3354	0,5000	VALIDATE	10	0,5321	0,3477
	Default Tree	Albero decisionale	0,5768	0,9033	0,0737	0,8090	5,1064	51,0638	0,5000	VALIDATE	10	0,5457	0,3943
	Entropy Tree	Albero decisionale	0,6530	0,9050	0,0696	0,8961	5,1698	51,6983	0,5000	VALIDATE	10	0,5478	0,3814
	Chi-Squared Tree	Albero decisionale	0,6134	0,9045	0,0712	0,8629	5,1171	51,1708	0,5000	VALIDATE	10	0,5393	0,3807

To prevent potential overfitting, various predictive modeling techniques provide a mechanism to adjust the complexity of the model. In the case of decision trees, this process is referred to as pruning.
The subtree method specifies the approach to construct the subtree using subtree methods. The following options are available:

- C4.5: Pruning is performed using the C4.5 algorithm.
- Cost complexity: The subtree with the lowest penalized average squared error (ASE) is selected.
- Reduced error: The smallest subtree with the best evaluation value is chosen.

We pruned our tree with all the three different options and the “Reduced Error” gave us the best KS value.
Even though the ‘Gini Tree’ was the best one at the beginning, after using the pruning option “reduced error” we found out that Entropy Tree gave us the best result possible. The entropy method for decision trees employs the information gain or decrease in entropy to divide each variable and determine the division.

Confronto di modelli

Champi...	Nome	Nome a...	KS (You...	Accurat...	Averag...	Area sot...	Lift cum...	Percent...	Cutoff	Ruolo d...	Profond...	Score F1	False Di...
★	Entropy Tree RE	Albero decisionale	0,7046	0,9091	0,0671	0,9086	5,2467	52,4672	0,5000	VALIDATE	10	0,5566	0,3518
	Gini Tree RE	Albero decisionale	0,6531	0,9075	0,0698	0,8953	5,1335	51,3354	0,5000	VALIDATE	10	0,5321	0,3477
	Default Tree	Albero decisionale	0,5768	0,9033	0,0737	0,8090	5,1064	51,0638	0,5000	VALIDATE	10	0,5457	0,3943
	Chi-Squared Tree RE	Albero decisionale	0,6489	0,9074	0,0700	0,8942	5,1292	51,2917	0,5000	VALIDATE	10	0,5306	0,3480
	Default Tree RE	Albero decisionale	0,5772	0,9039	0,0746	0,8074	4,8754	48,7539	0,5000	VALIDATE	10	0,4823	0,3473

GRADIENT BOOSTING

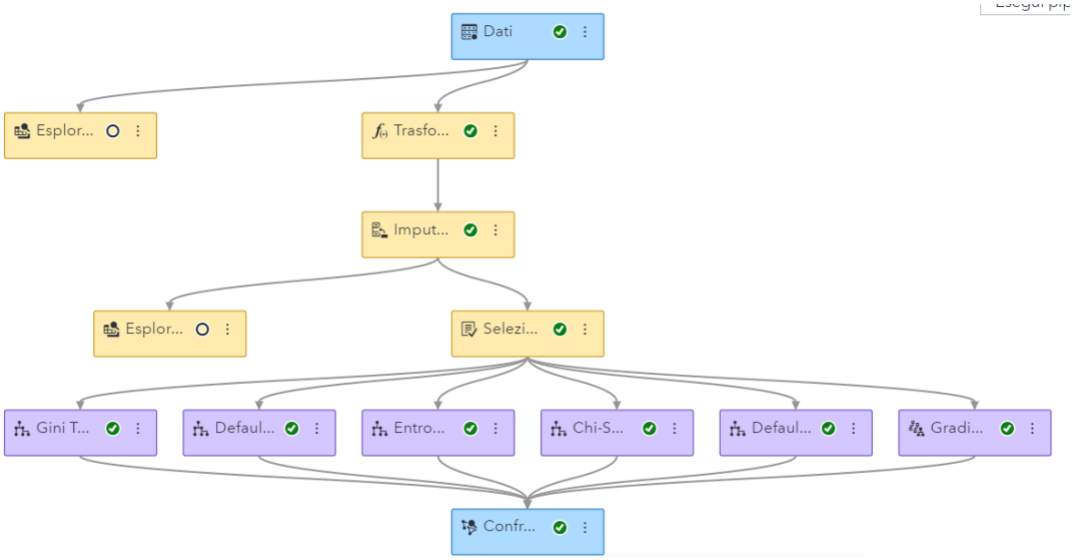
Boosting is an ensemble meta-algorithm in machine learning aimed at primarily reducing bias and, to some extent, variance. It has the capability to transform feeble learners into robust learners. By altering the training data in a sequential manner, it adapts to the outcomes of previous models. Instances that are misclassified receive greater emphasis in subsequent models.

The default settings for gradient boosting are: 100 trees, learning rate = 0,1 and categorical target metric is classification error.

We tried to modify the number of trees, the learning rate and the categorical target metric to 'Log Loss' but the default gradient boosting was still the best one.

Confronto di pipeline

Champion	Challenger	Nome	Nome algoritmo	Nome della pipeline	KS (Youden)	Num. osservazio
		Default Gradient boosting	Gradient boosting	Gradient Boosting	0,744	13.564
		Gradient boosting LL	Gradient boosting	Gradient Boosting	0,736	13.564
		Entropy Tree RE	Albero decisionale	Decision Tree	0,705	13.564
		Gini Tree RE	Albero decisionale	Decision Tree	0,653	13.564
		Chi-Squared Tree RE	Albero decisionale	Decision Tree	0,649	13.564
		Default Tree RE	Albero decisionale	Decision Tree	0,577	13.564
		Default Tree	Albero decisionale	Decision Tree	0,577	13.564



FOREST

A random forest is a collection of basic decision trees, each capable of generating its unique output based on a set of input variables. In classification tasks, this output takes the form of a class, which assigns a set of independent variables to one of the categories in the dependent variable.
A forest model comprises an indeterminate number of simple decision trees utilized to determine the ultimate result. In the case of a categorical target, the response from the ensemble of basic decision trees is the majority vote for the most popular class or the average of the posterior probabilities calculated by the individual trees. For a continuous target, the response from the ensemble model is the average estimate derived from the individual decision trees.

We tried different numbers of trees, changing the tree splitting criteria and other variables.
In the end, the best option was the Entropy Forest.
However, if compared to the models analyzed before, the Default Gradient Boosting is still the champion model.

Confronto di pipeline

Champion ↓	Challenger	Nome	Nome algoritmo	Nome della pipeline	KS (Youden)	Num. osservazio
★		Default Gradient boosting	Gradient boosting	Gradient Boosting	0,744	13.564
	📌	Gradient boosting LL	Gradient boosting	Gradient Boosting	0,736	13.564
		Entropy Forest	Forest	Forest	0,732	13.564
	📌	Default Forest	Forest	Forest	0,727	13.564
		Entropy Tree RE	Albero decisionale	Decision Tree	0,705	13.564
	📌	Gini Tree RE	Albero decisionale	Decision Tree	0,653	13.564
	📌	Chi-Squared Tree RE	Albero decisionale	Decision Tree	0,649	13.564
	📌	Default Tree RE	Albero decisionale	Decision Tree	0,577	13.564
	📌	Default Tree	Albero decisionale	Decision Tree	0,577	13.564

NEURAL NETWORK

Neural networks are a computational technique inspired by the structure and functioning of the human brain. They offer a versatile approach to approximating extremely nonlinear connections between variables, eliminating the necessity for preconceived assumptions about the nature of these relationships.

One significant advantage of neural networks lies in their boundless adaptability. Acting as universal approximators, they possess the capability to model any input-output association, regardless of its intricacy. Neural networks overcome the primary constraints encountered in traditional regression methods. However, they do have a couple of limitations of their own: limited interpretability and the requirement for substantial data signals.

We standardized the inputs and then we doubled the number of neurons in the hidden layer with respect to the number of input variables. We tried to add another layer but it led to worse results.
The table below shows the results of the Neural Network both on training and validation partitions.

Statistiche di bontà del modello

Nome t...	Ruolo d...	Indicato...	Partizio...	Num. os...	Averag...	Divisore...	Root Av...	Errore d...	Perdita l...	KS (You...	Area sot...	Coeffici...	Gamma
y	TRAIN	1	1	31.647	0,0718	31.647	0,2679	0,1023	0,2362	0,6545	0,9022	0,8043	0,8120
y	VALIDATE	0	0	13.564	0,0720	13.564	0,2683	0,1003	0,2372	0,6487	0,8990	0,7981	0,8058

MODEL COMPARISON and ASSESSMENT

The champion model of this project is the *Default Gradient Boosting*. The model was chosen based on the KS (Youden) for the validation partition (0,744).
Kolmogorov-Smirnov statistic (KS) is a goodness-of-fit index that represents the maximum separation between the model ROC curve and the baseline ROC curve.

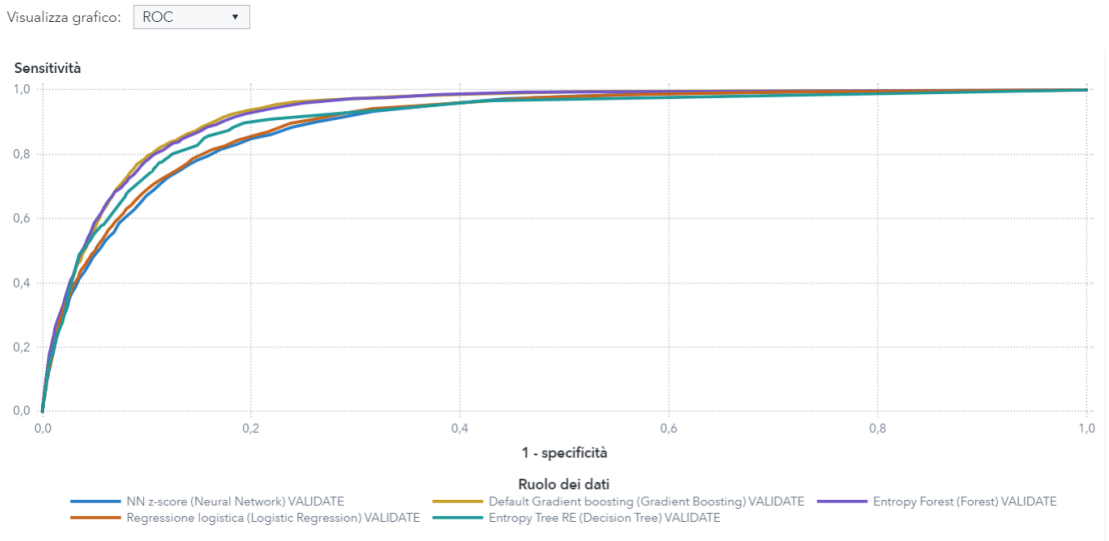
Confronto di pipeline

Champion	Nome	Nome algoritmo	Nome della pipeline	KS (Youden)	Num. osservazic
✓	Default Gradient boosting	Gradient boosting	Gradient Boosting	0,744	13.564
	Entropy Forest	Forest	Forest	0,732	13.564
	Entropy Tree RE	Albero decisionale	Decision Tree	0,705	13.564
	Regressione logistica	Regressione logistica	Logistic Regression	0,659	13.564
	NN z-score	Rete neurale	Neural Network	0,649	13.564

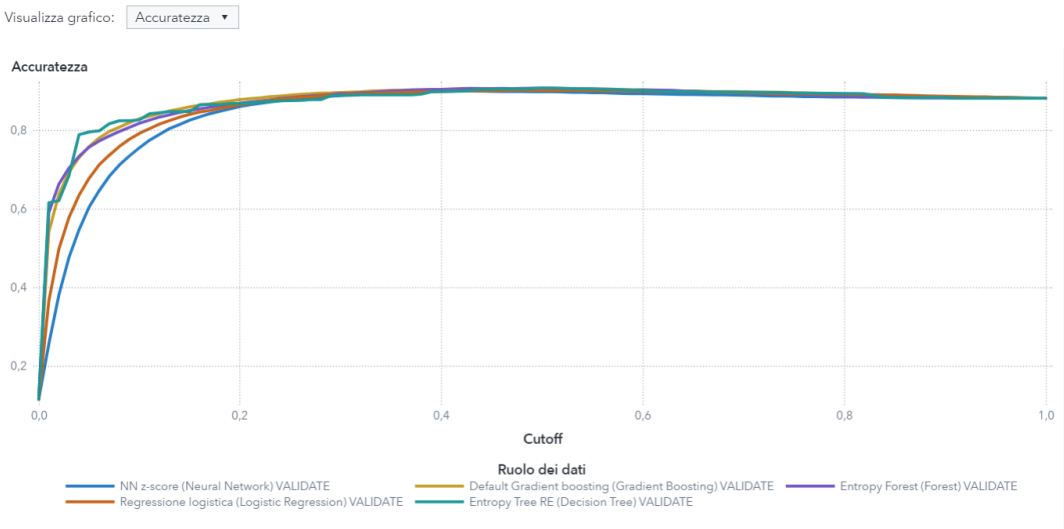
ROC analysis is a statistical method for evaluating the performance of a binary classifier. It plots the true positive rate (sensitivity) versus the false positive rate (1 - specificity) of a binary classifier for a range of threshold values.

90,57% of the validation partition was correctly classified using the Default Gradient Boosting model. The five most important factors are: 'Transformed duration', 'month', 'poutcome', 'day' and 'contact'.

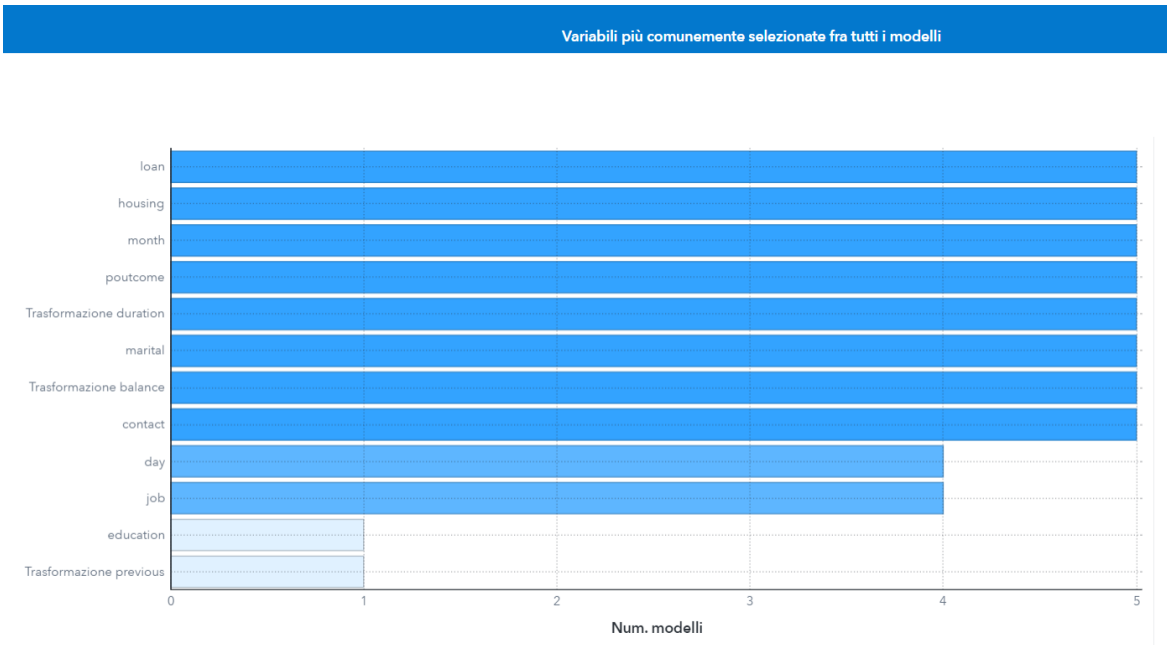
Model comparison ROC:



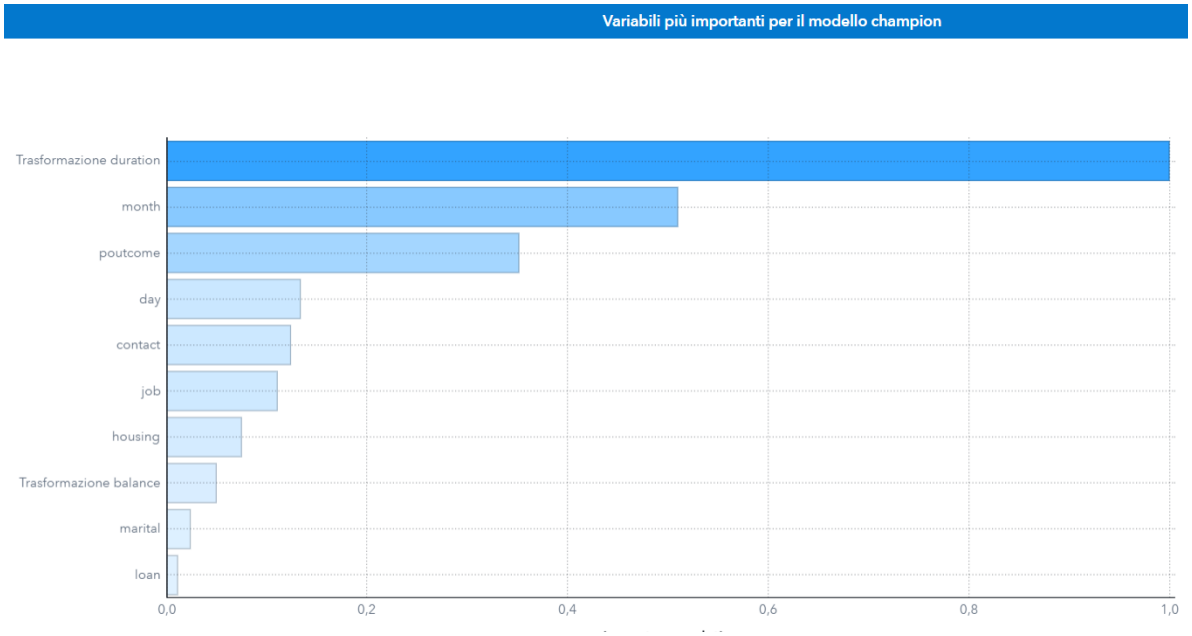
Model comparison Accuracy:



This plot shows the number of times that an input was deemed an important variable for any model that was used in Pipeline Comparison, including the pipeline champions and challenger models. Variable importance is calculated using a surrogate model, a one-level decision tree for each input where the target is the predicted class or value. Inputs with a positive importance value are determined to be important. The most important inputs across the champion and challenger models for this project appear at the top of the plot



This plot shows the 10 most important variables for the champion model (default gradient boosting), as determined by the relative importance calculated using the actual model. The most important input for this model is “transform duration”. The input “month” has a relative importance of 0.51, for example, which means it is 0.51 times as important as “transformed duration”.

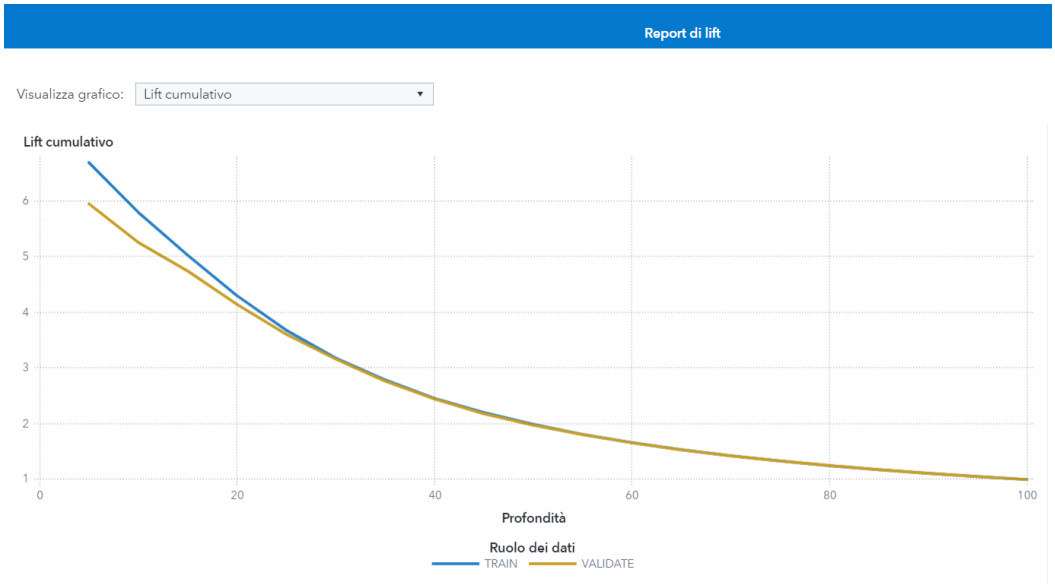


Default Gradient Boosting ASSESSMENT

We decided to assess the best model: Default Gradient Boosting.

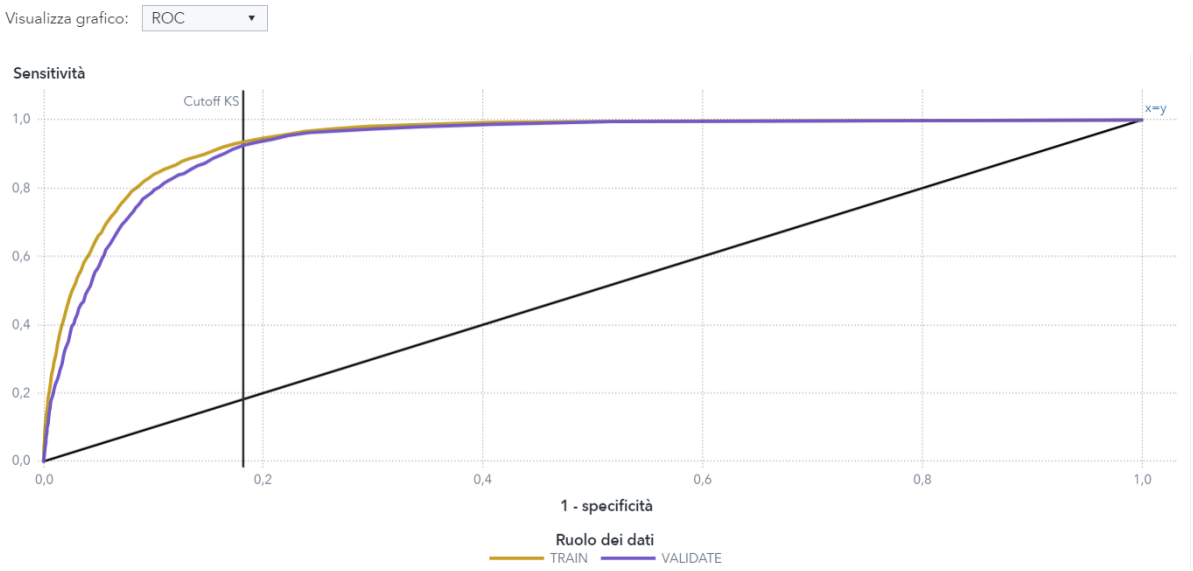
The Cumulative Lift of the validate partition is 5.26 at the 10% quantile, indicating that the first two quantiles have approximately 5.26 times more events than what would be expected by chance (10% of the total number of events). Since this value is above 1, it can be concluded that the use of our model to identify responders is more advantageous than not using a model, based on the selected partition.

Thus, cumulative lift measures how much more likely it is to observe an event in the quantiles than by selecting observations at random.



ROC analysis:
The true positive and false positive measures are calculated at various cutoff values. To help identify the best cutoff to use when scoring our data, the KS Cutoff reference line is drawn at the value of 1-specificity (false positive) where the greatest difference between true positive and false positive is observed for the validate partition. The KS Cutoff line is drawn at the cutoff value 0.1, where the value of 1-specificity is 0.182 and the sensitivity value is 0.926.

A ROC curve that rapidly approaches the upper-left corner of the graph, where the difference between true positive and false positive is the greatest, indicates a more accurate model. A diagonal line where they are equal indicates a random model.



Accuracy analysis:

For this model, the accuracy in the validate partition at the cutoff of 0.5 is 0.906.
Accuracy is the proportion of observations that are correctly classified as either an event or nonevent, calculated at various cutoff values. Accuracy is calculated as (true positives + true negatives) / (total observations).