The goal of the solution is the improve the sales of the call center. “Improving sales” is quite broad, however. Given that the campaign is time-consuming to call all customers, consumes a lot of call center resources, and there is a cost per minute for each call, the specific goal is to maximize the *expected profit* to the company using the call centers’ sales rather than maximizing the numbers of call. The business decision, therefore, is to decide which customer to call that would maximize the expected profit, which is achieved if the customer’s expected profit of calling them (EV(C) is greater than the value of not calling them, EV(NC).

We have the cost of the call is $1 per minute (denoted as m) while each insurance card purchased brings the value of $15. I created the following cost-benefit matrix.

	Buy (B)	Not buy (NB)
Call (C)	15-1*m = 15 - m	-1*m = -m
Not call (NC)	15	0

We want to call customers if the expected profit of calling them, (EV(C) is greater than the value of not calling them, EV(NC).
EV(C) = P(B|C)*(15-m) + P(NB|C)*(-m)
EV(NC) = P(B|NC)*15 + P(NB|NC)*0 = P(B|NC)*15

We want to call the customers if EV(C) - EV(NC) > 0:
P(B|C)*(15-m) + P(NB|C)*(-m) - P(B|NC)*15 > 0
P(B|C)*(15-m) + ((1-P(B|C))*(-m) - P(B|NC)*15 > 0
P(B|C)*(15) + P(B|C)*(-m) + (-m) + P(B|C)*(m) - P(B|NC)*15 > 0
P(B|C)*(15) + (-m) - P(B|NC)*15 > 0
[P(B|C)-P(B|NC)]*15 - m > 0

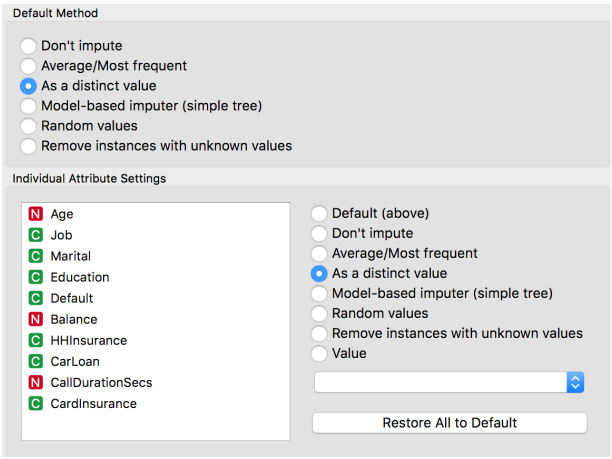
Ideally, we would like to build two models to predict P(B|C) and P(B|NC). However, we only have data on customers who we already called so we are unable to build a predictive model for P(B|NC). For now, we will assume the worst-case scenario that P(B|NC) = 0%, meaning that if IBB does not call the customer, they would not be aware of the product and therefore, would not purchase. Thus, we would call a customer if their P(B|C)*15 + (-m) > 0 and build a model to predict P(B|C).

The instance is a customer of IBB, identified by their ID (meta attribute). The target variable is whether or not the customer buys the credit card insurance. The features are age (numeric), job (categorical), marital status (categorical), education (categorical), default (categorical), balance (numeric), HHInsurance (categorical), carloan (categorical), communication (categorical), and call duration (numeric). To double-check that these features provide information to the model and are not redundant, I examined the Feature Statistics.

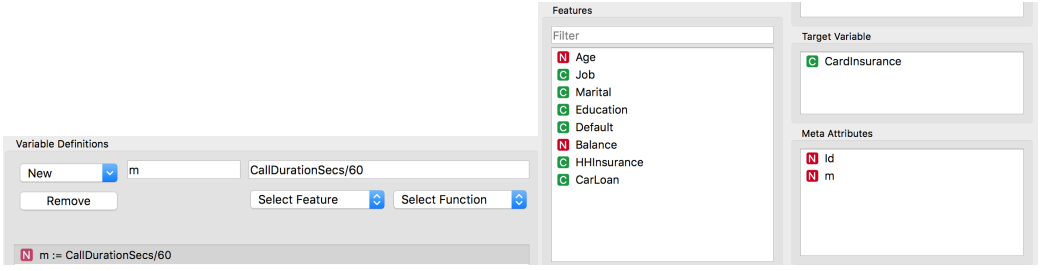
	Name	Type	Role	Values
1	**Id**	**numeric**	**meta**	
2	Age	numeric	feature	
3	Job	categorical	feature	admin., blue-collar, entrepreneur, ho...
4	Marital	categorical	feature	divorced, married, single
5	Education	categorical	feature	primary, secondary, tertiary
6	Default	categorical	feature	0, 1
7	Balance	numeric	feature	
8	HHInsurance	categorical	feature	0, 1
9	CarLoan	categorical	feature	0, 1
10	Communicati...	categorical	feature	cellular, other, telephone
11	CallDuration...	numeric	feature	
12	**CardInsura...**	**categori...**	**target**	**0, 1**

Name	Distribution	Center	Dispersion	Min.	Max.
Id		1750.50	0.58	1.00	350
CallDuration...		350.57	0.98	5.00	325
Balance		1541.77	2.34	-3058.00	984
Age		41.23	0.28	18.00	9
CardInsurance		0	0.67		
Communicati...		cellular	0.76		
CarLoan		0	0.40		
HHInsurance		0	0.69		
Default		0	0.08		
Education		secondary	0.99		
Marital		married	0.94		
Job		management	2.12		

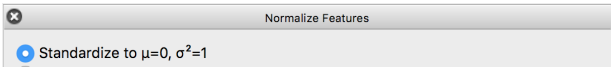
Next, I noticed that there are 0.4% missing values, mostly from education and job. These values missing could imply that a customer does not have an educational background or does not have a job, and therefore, should not be ignored. I impute missing values as distinct values.



I created another variable, m, which represent the call duration in minutes and takes the value of CallDurationSecs/12. I set the value as a meta attribute so that is it not normalized by preprocessing since I would use the value to calculate costs later. I removed CallDurationSecs because we will not have that information for future data to predict whether or not we should call the customer.

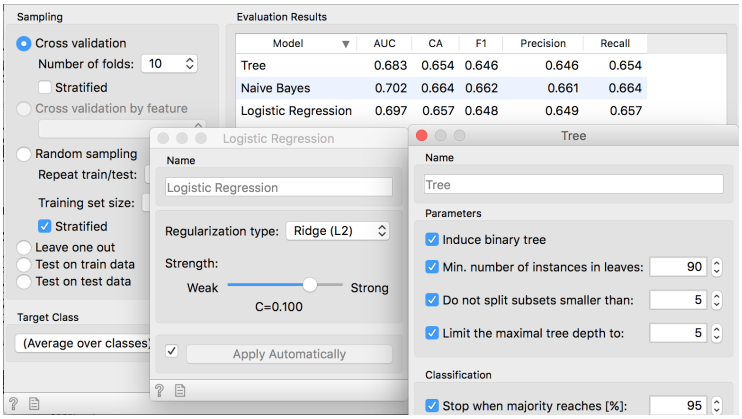


I then preprocessed the data, standardizing the numeric features with a mean of 0 and variance of 1.

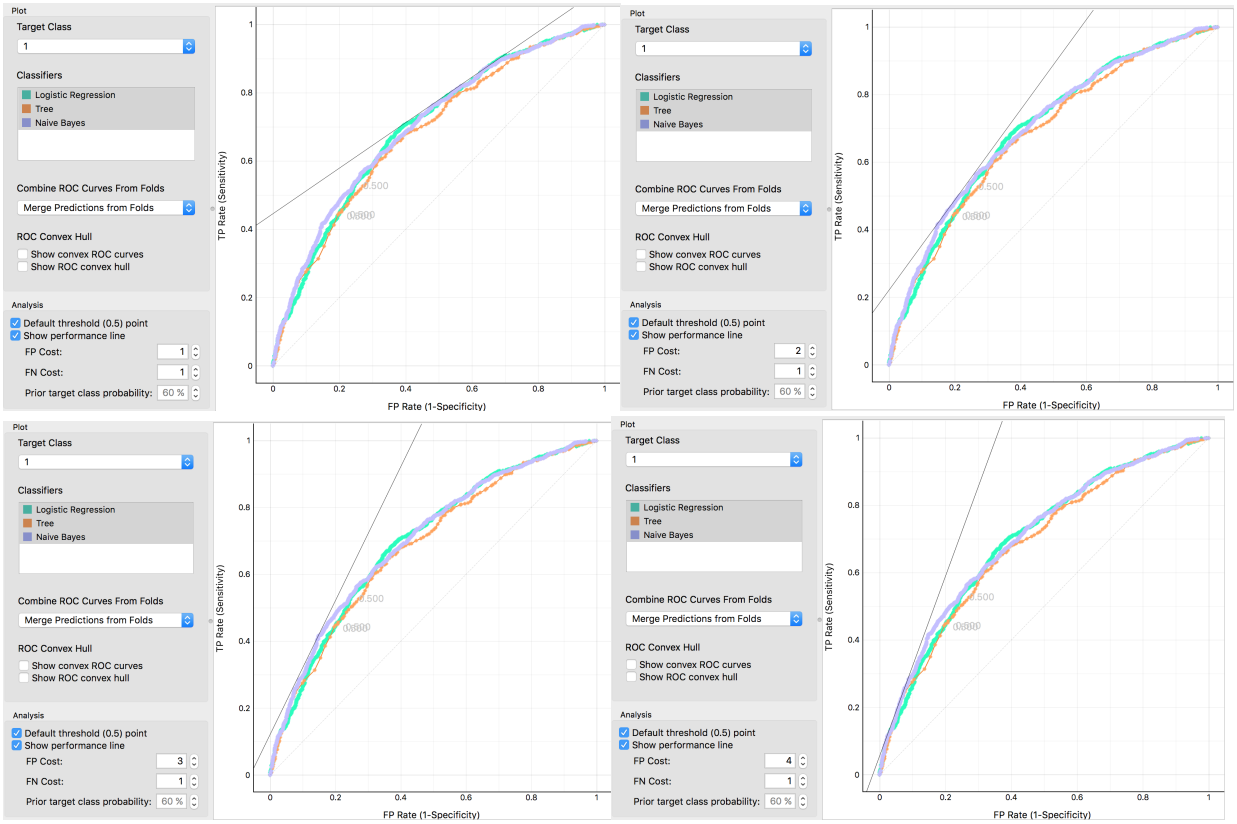


I sampled 80% of the dataset to use for training the models, and the remaining 20% to test the model. I selected replicable sampling to maintain consistent answers. I used cross-validation with 10 folds to compare models with AUC. I tested the model with three algorithms, specifically, decision

tree, logistic regression and Naive Bayes. I adjusted the tree minimum number of instances to find where it best performs. I tested the tree model with the min. number of instances ranging from 1 to 150. The optimal number is 90 instances and tree dept of 5, yielding an AUC of 0.683. For the logistic regression model, I tried both Ridge and Lasso regularization with complexity ranging from 500 to 0.013. The max AUC of the logistic regression model of 0.697 is at C=0.1 and Ridge regularization. Naive Bayes ultimately outperforms the other algorithms with an AUC of 0.702. Since the AUC performance are quite close, I examined further with ROC analysis.



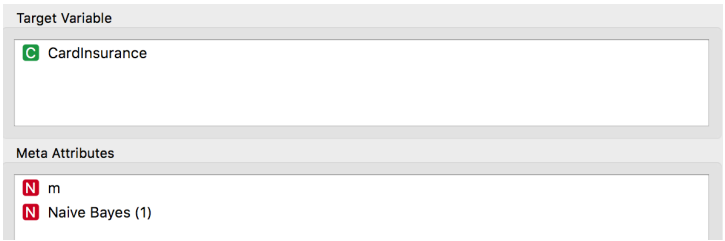
In the ROC analysis, the three models closely follow each other for the most part. Naive Bayes mostly outperforms decision tree and LR except for some points, specifically when FN = FP.



However, realistically speaking, such cases are not possible. FP occurs when we think that a customer would purchase, and thus call them when they actually don't purchase -- the cost of such

error is -1*duration of the call in minutes. According to historical data, the call can range from anywhere between a minute to around 25 minutes, making the cost range from -$1 to -$25. On the other hand, FN occurs when we think that a customer would not purchase, and therefore, not call them when they actually purchase. This error is of no cost to us, and rather, give us a profit of $15, which is the price of an insurance card. Thus, there is no possible case where FN equals FP. I focused on the models' performances when FP is more costly than FN, which is more realistic given the aforementioned analysis. Since FP has a range of costs, I examined the ROC chart at multiple points. Using the iso performance line, we can see that Naive Bayes outperforms the other models when FP cost is higher than FN cost. Therefore, I decided to proceed with the Naive Bayes.

I obtain the model's output on the test data by linking the Naive Bayes widget and the remaining data from data sampler to prediction. I exported the following variables to calculate the potential economic impact of the solution.



According to the model, the company should have called 415 instead of calling everyone. The economic impact, compared to the base value of calling everyone is $1614.96. The impact per decision is $2.31 while the revenue per decision is $2.60. Given that there are 30,000 customers remaining, the potential revenue from those customers is $77,975.

CardInsurai	Naive Baye	m	EV©-EV(NC	Call	Revenue(M	Base Revenue (Calling	Impact
0	0.759264	2.666667	$ 8.72	1	8.722291	-2.666666667	11.38896
1	0.596145	4.033333	$ 4.91	1	0.466667	10.96666667	-10.5
0	0.496803	2.633333	$ 4.82	1	4.818718	-2.633333333	7.452051
0	0.402668	6.016667	$ 0.02	1	0.023353	-6.016666667	6.04002
1	0.722798	3.15	$ 7.69	1	1.35	11.85	-10.5
0	0.161963	0.883333	$ 1.55	1	1.546114	-0.883333333	2.429447
1	0.828038	22.43333	$ (10.01)	0	4.5	-7.433333333	11.93333
1	0.268919	5.083333	$ (1.05)	0	4.5	9.916666667	-5.41667
1	0.130069	13.85	$ (11.90)	0	4.5	1.15	3.35
1	0.740156	8.416667	$ 2.69	1	-3.91667	6.583333333	-10.5
0	0.26226	1.866667	$ 2.07	1	2.067235	-1.866666667	3.933902
0	0.122213	3.433333	$ (1.60)	0	0	-3.433333333	3.433333
0	0.431724	5.6	$ 0.88	1	0.875864	-5.6	6.475864
0	0.09729	1.033333	$ 0.43	1	0.426016	-1.033333333	1.45935
0	0.212793	6.383333	$ (3.19)	0	0	-6.383333333	6.383333
1	0.2906	22.55	$ (18.19)	0	4.5	-7.55	12.05
1	0.662363	12.41667	$ (2.48)	0	4.5	2.583333333	1.916667
1	0.774412	7.583333	$ 4.03	1	-3.08333	7.416666667	-10.5

Target decision: EV©-EV(NC) > 0		
Major Assumption: P(B|NC)		
0%		
Reality: P(B|NC)		
30%		
Profit per Card	$	15.00
Model Revenue	$	1,819.43
Base Revenue	$	204.47
Economic Impact	$	1,614.96
Increase in Revenue		790%
# of People		700
# of People Called		415
Impact per Decision	$	2.31
Revenue per Decision	$	2.60
People Remaining		30000
Potential Impact	$	69,212.66
Potential Revenue	$	77,975.52

I'd like to walk through the revenue calculation since it is rather complicated and holds a few assumptions. In evaluation, instead of using the assumed value of P(BINC) = 0% as mentioned previously, I used the real probability. Given the aforementioned confusion matrix, we have the following scenarios.

Scenario 1:

Assumptions:

1. if the customer already bought and we call, there is a chance that the customer could ask for a refund: P(NCIC) is not 0% and the value would be -15-m

2. If the customer already bought and we call, there is a chance that the customer could buy another insurance card. That probability is P(B|C) with value of 15-m

Therefore, if the customer bought then the value would be calculated as:
P(B|NC)*15 + C[(P(B|C)*(15-m)+P(NB|C)*(-15-m)] = P(B|NC)*15 + C*[2P(B|C)(15)-15-m]

This is not totally realistic as one cannot simply just return insurance. I proceeded with the following scenario.

Scenario 2:

Assumptions:

1. If the customer already bought and we call, there is no chance that they would ask for a refund and no longer use the insurance: P(NB|C) = 0
2. If the customer already bought and we called, then they for sure will still continue with the insurance product: P(B|C) = 1 with value = -m

Therefore, if the customer bought the insurance card, the value would be: P(B|NC)*15+C*(-m)

If the customer did not buy, then we have the following value calculation:
0+C[P(B|C)(15-m)+P(NB|C)*(-m) = C*[P(B|C)*15-m]