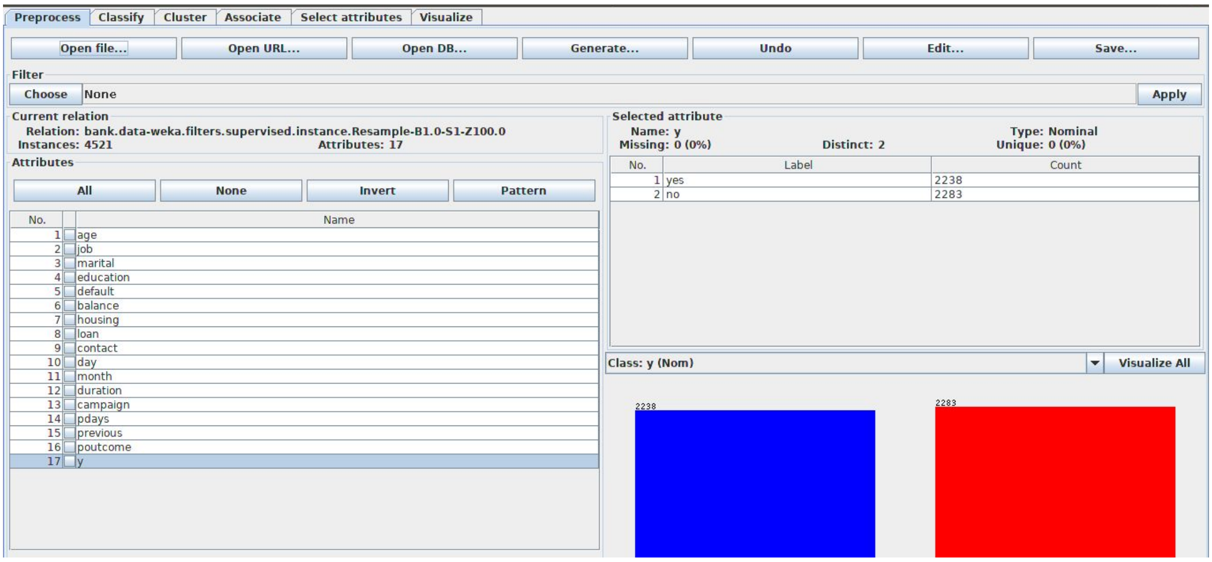
173705
172808

Direct Marketing Strategy for Gloucester's Bank

In order to proceed with the study you claimed to us, we have used a machine learning's suit called Weka in order to achieve your order.
For the sake of determining how good will be our suggested model for your campaign, we have taken the more basic classifier in order to contrast how good is our proposal.



Current relation
Relation: bank.data-weka.filters.supervised.instance.Resample-B1.0-S1-Z100.0
Instances: 4521 Attributes: 17

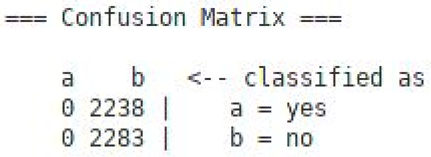
Attributes

No.	Name
1	age
2	job
3	marital
4	education
5	default
6	balance
7	housing
8	loan
9	contact
10	day
11	month
12	duration
13	campaign
14	pdays
15	previous
16	poutcome
17	y

Selected attribute
Name: y Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count
1	yes	2238
2	no	2283

As far as we know, the classifier that will choose the most common class among all the 16 attributes you are giving to us, corresponds to ZeroR Classifier, considering that it will ignore all the input attributes and predict the most common class based on the output value. So as you can see in the right-up corner table, we have 2238 classes predicted as "Yes" and 2283 predicted as "No", so we will expect approximately a 50% of classification accuracy.
After running the classifier, we have obtained the following values:



```
=== Confusion Matrix ===

   a    b   <-- classified as
   0 2238 |    a = yes
   0 2283 |    b = no
```

On the basis of the results and the confusion matrix, we can calculate also manually the accuracy by applying the formula:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

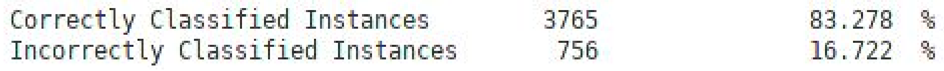
$$Accuracy \; = \frac{0 + 2283}{0 + 2283 + 0 + 2238} \; = 0,504977$$

So, this result will be useful in order to determine the performance for other classification methods.

Here you have some examples of how good are the models we have tested, in terms of accuracy.

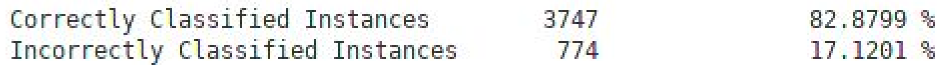
Logistic Regression:

- Using training set:



```
Correctly Classified Instances        3765               83.278  %
Incorrectly Classified Instances       756               16.722  %
```

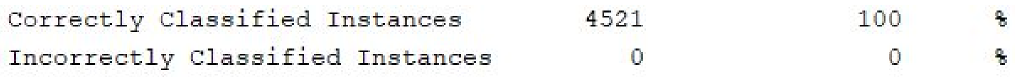
- Using 10-fold cross validation:



```
Correctly Classified Instances        3747               82.8799 %
Incorrectly Classified Instances       774               17.1201 %
```

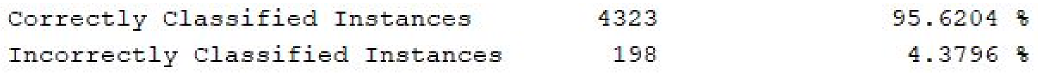
K-NN:

- Using training set:



```
Correctly Classified Instances        4521               100      %
Incorrectly Classified Instances         0                 0      %
```

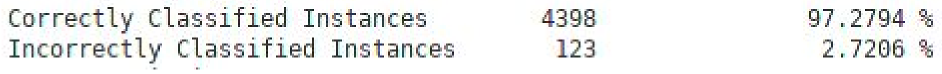
- Using 10-fold cross validation:



```
Correctly Classified Instances        4323               95.6204 %
Incorrectly Classified Instances       198                4.3796 %
```

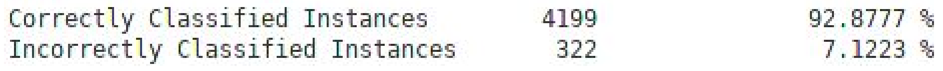
DECISION TREES:

- Using training set:



```
Correctly Classified Instances        4398               97.2794 %
Incorrectly Classified Instances       123                2.7206 %
```

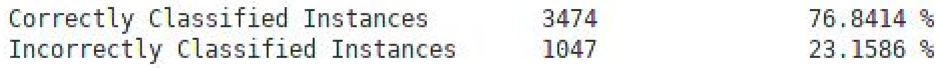
- Using 10-fold cross validation:



```
Correctly Classified Instances        4199               92.8777 %
Incorrectly Classified Instances       322                7.1223 %
```

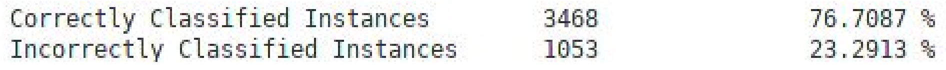
NAIVE-BAYES:

- Using training set:



```
Correctly Classified Instances        3474               76.8414 %
Incorrectly Classified Instances      1047               23.1586 %
```

- Using 10-fold cross validation:

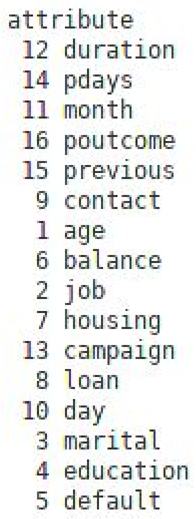


```
Correctly Classified Instances        3468               76.7087 %
Incorrectly Classified Instances      1053               23.2913 %
```

For your understanding, we have excluded all those models which were generated from a training set, since its results will not fit a real situation, because we will always obtain better accuracies when learning from a unique dataset which is learning from itself.

Having in mind this, what we want is separate data; a part for training and a part of testing. This behaviour can be achieved by using a 10-folds Cross Validation. Due to this, we will obtain a more averaged results and as consequence, we will work in a more realistic scenario.
By seeing the numbers, we have seen that K-NN is the algorithm with the best accuracy for the characteristics of our dataset. Since we have a not very large dataset, we want to predict a binary class and we have labeled data, our reasoning at this point is that we have to take into account that we have more nominal attributes than numerical. For this reason, the most accurate classifying algorithm should be Naive-Bayes but in the domain of our problem we have seen with the previous results that the K-NN classifying algorithm has more accuracy. This allows us to think that the numerical values are more relevant attributes than the nominal ones.

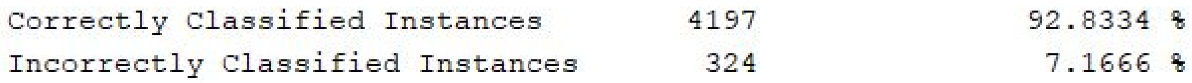
After having done a 10-cross validation Information Gain feature selection, we have seen that the most important attributes are the following ones:



```
attribute
 12 duration
 14 pdays
 11 month
 16 poutcome
 15 previous
  9 contact
  1 age
  6 balance
  2 job
  7 housing
 13 campaign
  8 loan
 10 day
  3 marital
  4 education
  5 default
```

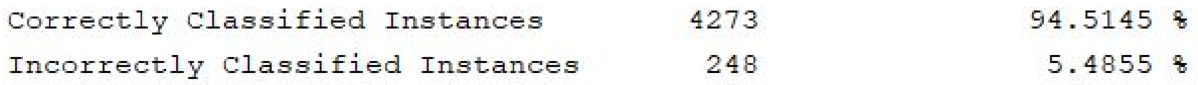
As we deduced previously, we have in the first five positions one more numerical attribute than nominal. In order to check this fact, we have run the k-NN algorithm twice: one time only with the 5 more relevant attributes, and the other with the remaining attributes. The results show that we have less accuracy with 11 attributes (less relevant) than with 5 (most relevant), so we can confirm that those 5 attributes are the most relevant ones as we said previously:

First case (11 less relevant attributes):



```
Correctly Classified Instances        4197               92.8334 %
Incorrectly Classified Instances       324                7.1666 %
```

Second case (5 most relevant attributes):



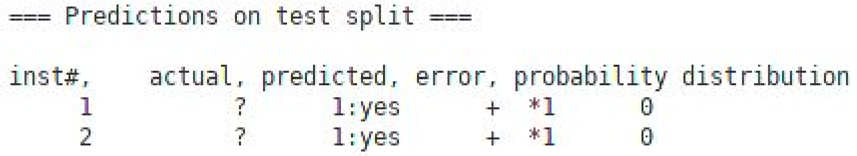
```
Correctly Classified Instances        4273               94.5145 %
Incorrectly Classified Instances       248                5.4855 %
```

Taking the following values for two people characteristics, we have generate a .arff file in order to see how it classifies the samples.

- **Person1:** 66, retired, married, primary, no, 206, no, no, cellular, 9, feb, 479, 1, -1, 0, unknown

- **Person2:** 54, technician, married, tertiary, no, 876, no, no, cellular, 27, oct, 269, 3, 541, 3, success

So as we have seen on Weka, after loading test.arff, which contains the users to be tested, we can classify these two instances as potential clients for the bank.



```
=== Predictions on test split ===

inst#,    actual, predicted, error, probability distribution
     1         ?      1:yes      +   *1      0
     2         ?      1:yes      +   *1      0
```

Having these results, we have to recall to the most common attributes to take a final decision. We have elaborated this board in order to visualize better what of the most common characteristics could influentiate the clients to come to your bank.

Attribute	Input Person 1	Input Person 2
Month	February	October
Duration	479	269
Previous Days	-1	541
Previous	0	3
Poutcome	Unknown	Success

Our conclusion is that in order to attract more clients the best months to release the campaign are in the coldest and with a contact superior to a threshold of 250 seconds, so these are the most important attributes to take into account. By the other hand, neither the previous contacts with the clients, the previous number of contacts performed a campaign nor the previous marketing campaign's outputs are as relevant as the month and duration are. The model we suggest you to use is the k nearest neighbors algorithm.