|  |  |
| --- | --- |
| **P170M109 Computational Intelligence and Decision** | **Date: 30 May 2018** |
| TOPIC 6 | **Students: Erika Gardini – Mattia Fucili** |

Provide the results obtained following the steps given in the lab work description (txt file). Please, plan your time to meet deadlines. This document (report) should be uploaded to Moodle system.

1. *Choose the error-based learning model suitable for your dataset*

Since our dataset contains a categorical feature as target feature, the model we have chosen is logistic regression.

1. *Implement cross-validation by selecting at least 4 folds*

First of all, we have created 4 different datasets shuffling the rows of the starting one using this code

credit1 <- credit[sample(nrow(credit)),]

credit2 <- credit[sample(nrow(credit)),]

credit3 <- credit[sample(nrow(credit)),]

credit4 <- credit[sample(nrow(credit)),]

To obtain the two datasets (training and test) we used this piece of code:

samp\_size\_1 <- floor(0.1 \* nrow(credit1))

set.seed(123)

train\_ind\_1 <- sample(seq\_len(nrow(credit1)), size = samp\_size\_1)

train\_set\_1 <- credit1[train\_ind\_1, ]

test\_set\_1 <- credit1[-train\_ind\_1, ]

The same for each new pair train-test.

Here the results of this split:

The proportion of the class variable in both sets is quite similar, in fact the class “Good” in the training set is the 69.55% of the whole dataset and in the test set is 74% and the class “Bad” in the training set is the 30.44% of the whole dataset and in the test set is 26%.

1. *Experiment: Select at least two categorical features and at least two continuous features. Perform necessary steps before learning it. Learn the model and write down its formal description (equation). Provide predictions for your target feature using test dataset.*

For this experiment we have split our categorical features in as many columns as the cardinality of the feature we are focus in. These new columns contain 1 or 0 depending on the value assumed by the original feature. To feed the continuous columns feature we have normalized them in the interval [-1; 1].

After this step we have computed the linear model as follows:

glm.fit <- glm(Class ~ inst4 + num1 + Age + Amount, data = train, family = binomial)

The resulting formulas of the model using four training sets are:

1. Training set 1
2. Training set 2
3. Training set 3
4. Training set 4

Applying it to test sets we have obtained the following confusion matrices:

1. Test set 1

|  |  |  |
| --- | --- | --- |
|  | Good | Bad |
| Good | 13 | 15 |
| Bad | 18 | 54 |

1. Test set 2

|  |  |  |
| --- | --- | --- |
|  | Good | Bad |
| Good | 9 | 13 |
| Bad | 19 | 59 |

1. Test set 3

|  |  |  |
| --- | --- | --- |
|  | Good | Bad |
| Good | 11 | 11 |
| Bad | 20 | 58 |

1. Test set 4

|  |  |  |
| --- | --- | --- |
|  | Good | Bad |
| Good | 8 | 10 |
| Bad | 19 | 63 |

Every result has been obtained calculating the perfect threshold by hand, since the library gives a value that makes bad results. In particular the threshold value has been tuned in such a way to achieve the best accuracy and trying to keep the false negative and false positive rate as lower as possible.

1. *Experiment: Select at least 6 features OR all features you have in training dataset. Perform necessary steps before learning it. Learn the model and write down its formal description (equation). Provide predictions for your target feature using test dataset.*

For this experiment we have used the same features transformed in the previous point.

We have computed the linear model as follows:

glm.fit <- glm(Class ~ inst4 + num1 + res4 + res2 + Age + Amount, data = train, family = binomial)

The resulting formulas of the model using four training sets are:

1. Training set 1
2. Training set 2
3. Training set 3
4. Training set 4

Applying it to test sets we have obtained the following confusion matrices:

1. Test set 1

|  |  |  |
| --- | --- | --- |
|  | Good | Bad |
| Good | 8 | 11 |
| Bad | 23 | 58 |

1. Test set 2

|  |  |  |
| --- | --- | --- |
|  | Good | Bad |
| Good | 7 | 9 |
| Bad | 21 | 63 |

1. Test set 3

|  |  |  |
| --- | --- | --- |
|  | Good | Bad |
| Good | 6 | 5 |
| Bad | 25 | 64 |

1. Test set 4

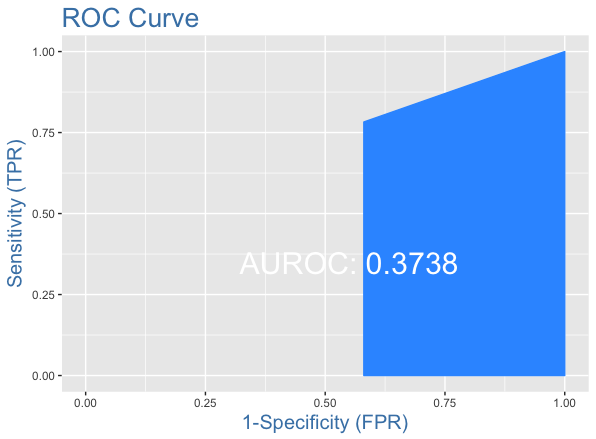
|  |  |  |
| --- | --- | --- |
|  | Good | Bad |
| Good | 8 | 9 |
| Bad | 19 | 64 |

Every result has been obtained calculating the perfect threshold by hand, since the library gives a value that makes bad results. In particular the threshold value has been tuned in such a way to achieve the best accuracy and trying to keep the false negative and false positive rate as lower as possible.

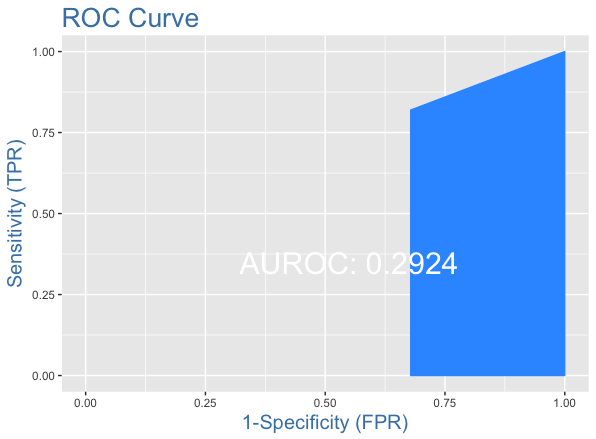
1. *For each experiment, find F1 score, provide ROC (AUC) curve and make conclusions about the goodness of model to be learned.*

Results of the experiment 3:

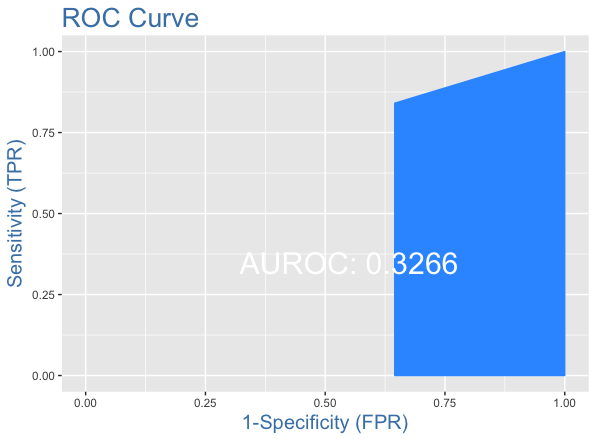
1. Test set 1



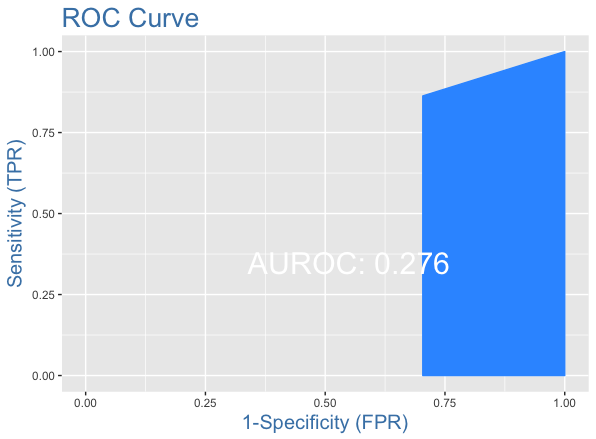
1. Test set 2



1. Test set 3

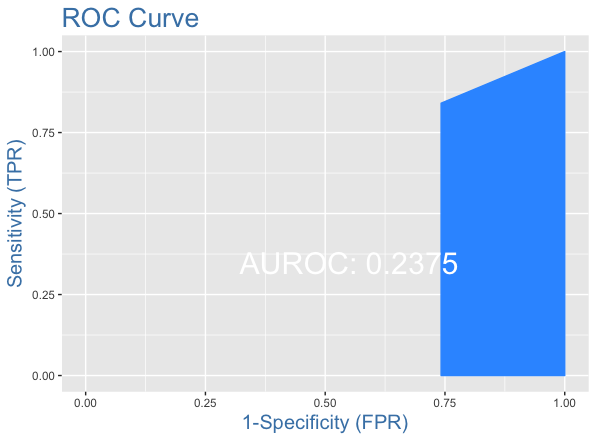


1. Test set 4

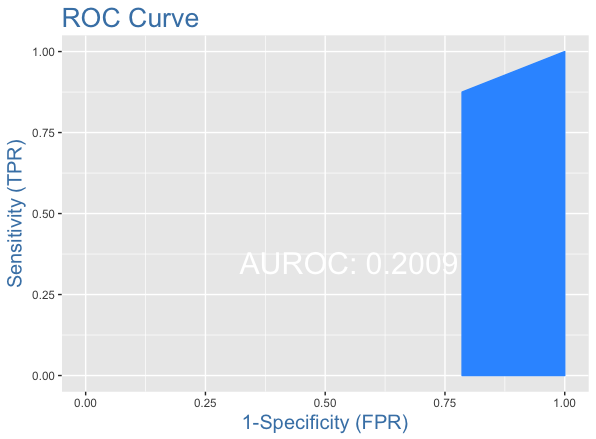


Results of the experiment 4:

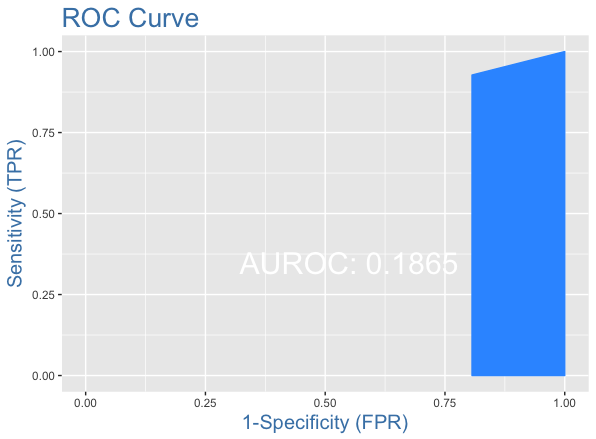
1. Test set 1



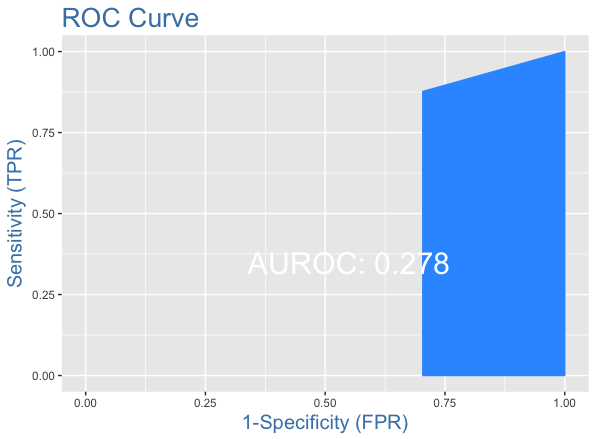
1. Test set 2



1. Test set 3



1. Test set 4



As we can expect from the results obtained in the previous two points the F1 scores of our logistic regressor classifier are very low, that means bad classification. Moreover, we can also say that looking at graphs of ROC curve and AUC.

1. *Summarize the results to be obtained.*

After a plenty of machine learning techniques applied in this credit dataset we can say that neither the other methods nor this logistic regressor could fit a good model to predict the class value of a specific bank creditor.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *Algorithm KNN* | | *Algorithm C5.0* | *Algorithm naïve Bayes* | *Logistic Regressor* |
|  | *Max percentage of correctly classified* | *Min percentage of correctly classified* | *Max percentage of correctly classified* | *Max percentage of correctly classified* | *Max percentage of correctly classified* |
| Categorical features | 70% | 64% | - | 75% | - |
| Numerical features | 68% | 60% | - | 74% | - |
| Categorical and numerical features | 74% | 62% | 64% | 72% | 71% |
| Categorical, numerical and derived features | - | - | 75% | - | - |
| All features | - | - | 70% | - | - |

As we can see from the above table the performance of this classifier, in terms of correct classification, goes even worst respect of the other classifiers.