Statistical and Machine Learning Approaches

Individual Assignment – Eduardo Razo

This document aims to describe and explain several Machine Learning approaches used in the project and to illustrate the different methodologies through the application of them on a real-life case. In this case, predicting the probability of a client to subscribe or not based on marketing information from a financial institution.

Part 1 – Machine Learning Algorithms

As a starting point, the approaches are described as follow:

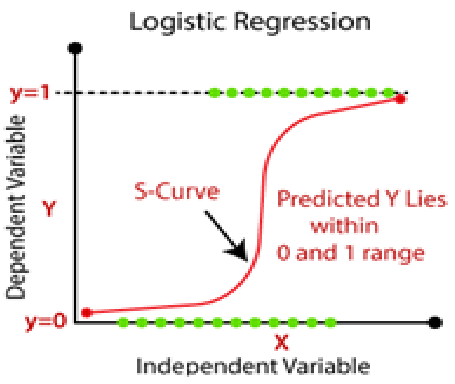
Logistic Regression

It is one of the most popular Machine Learning algorithms which come under supervised learning technique. We leverage this methodology when the target variable is categorical, and it is necessary to solve a classification task. It is used to predict the dependent variable based on the independent variables contained in a dataset. The output of the logistic regression should be categorical such as 0 or 1, Yes or No.

For example: Whether the stock market is going to go up (1) or down (0).

It is worthy to mention that the logistic regression is a linear method, but the predictions are transformed using the sigmoid function, which is a mathematical function that has a characteristic "S"-shaped.

Logistic regression becomes a classification technique only when a decision threshold is set. The setting of the threshold value is a very important aspect of logistic regression. The decision for the value of the threshold value is majorly affected by the values of precision and recall. Ideally, we look for precision and recall to be 1.



Logistic function

$$f(x) = \frac{1}{1+e^{-x}}$$

The parameters of a logistic regression model can be estimated by the probabilistic framework called maximum likelihood. Under this framework, a probability distribution for the target variable must be assumed and then a likelihood function defined that calculates the probability of observing the outcome given the input data and the model. This function can then be optimized to find the set of parameters that results in the largest sum likelihood over the training dataset.

The model is defined in terms of parameters called coefficients (*beta*), where there is one coefficient per input and an additional coefficient that provides the intercept or bias.

Advantages	Disadvantages
• Easy to implement, interpret and verify efficient to train.	• Assumption of lenearity between the dependant and the independent variable (rarely linear in real world).
• Less prone to overfitting but it can overfit in high dimensional datasets.	• Tends to overfit when the number of observations is less than the number of features.

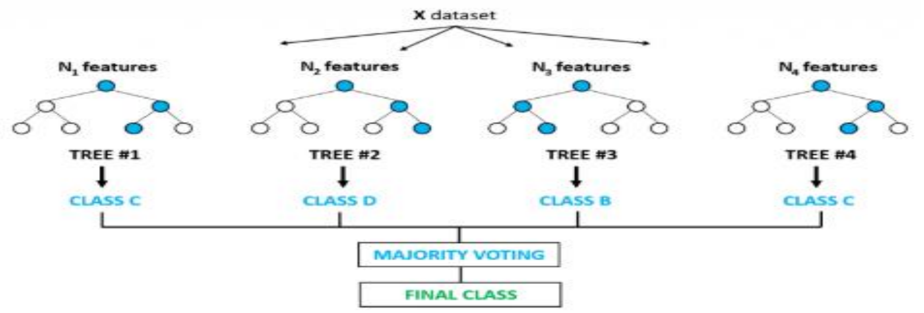
Random Forest

It is a supervised method based on an ensemble of decision trees that can be used for classification and regression tasks. Most of the times, this model is trained with the bagging method, in which the general idea is the combination of learning models that increases the overall result. It is an estimator that fits several decision trees on several sub samples of the data. On each iteration, the algorithm uses averaging of the error (which is the objective function) to improve the predictive accuracy and to manage overfitting.

Random forest grows many classification trees and each tree gives a classification, so that we can say that the tree votes for that specific class. The forest selects a classification depending on the most votes on each tree within the forest.

Trees grown as follow:

- Considering that there are **M** variables, a number **m** is specified at each node; **m** variables are selected randomly out of the **M**, and the best split **(m)** is used to split the node. The value of **m** is held during the forest growing.
- Each forest is grown to the largest possible. There is no pruning.



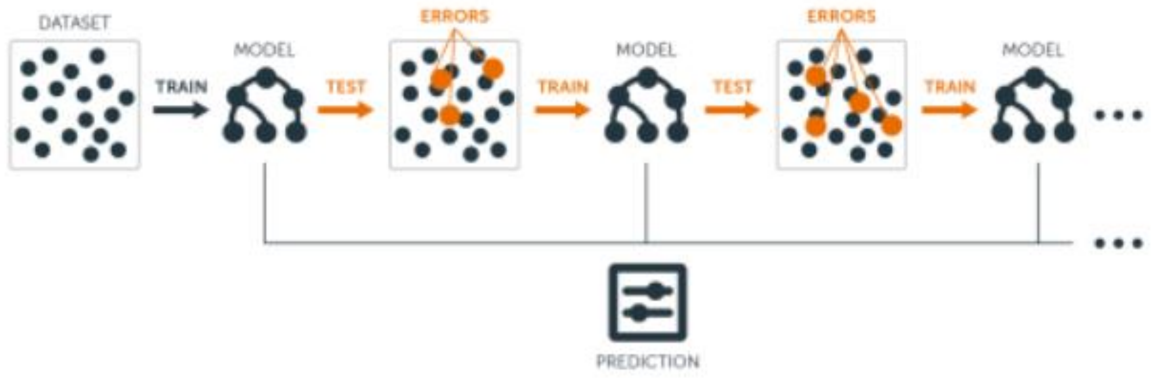
Advantages	Disadvantages
• Flexible and easy to implement.	• Training a large number of deep trees can have high computational costs.
• The algorithm can be used for classifiaction and regression.	• Requires much time to train in comparison to decision trees.
• Provides reliable feature importance estiamtion.	

Gradient Boosting Machine (GBM)

Gradient boosting is also a popular model can be leveraged for regression and classification problems.

Starting from the premise of boosting. Boosting is a method that consists in converting weak learners into strong ones. The main concept of this method is to improve (boost) the week learners sequentially and increase the model accuracy with a combined model. The algorithm trains many models in a sequential and additive way and performs by using gradients in the loss function. Regarding to the loss function, this is the metric used to determine how efficient is the model at fitting the data, in this case, the error or residual.

In this case, decision trees are typically the weak learners in gradient boosting.

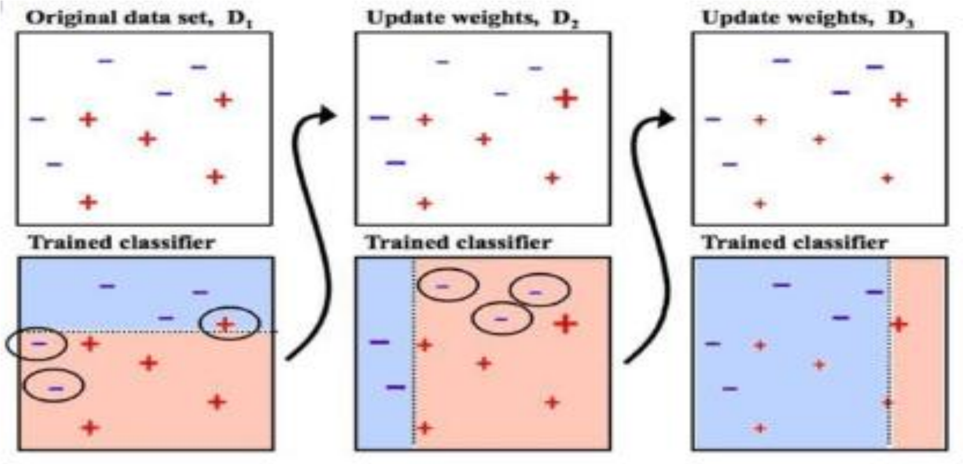


Advantages	Disadvantages
• Easy to read and interpret, making predictions easy to handle.	• The methos is sensiteve to outliers.
• It is a method that can mitigate overfitting.	• Training is time consuming.

Adaptive Boosting (Adaboost)

The objective of this method as well as Gradient Boosting, is to combine several weak learners to conform a stronger learner. It also focuses on weak learners, which are usually decision trees. During the fitting process, previous errors are corrected and any observation that was misclassified are assigned with more weight than others that were classified correctly. Taking into consideration this general idea, the prior error is adjusted in each model through weighting until an accurate prediction is made.

The models are created one after the other, each model updating the weights on the training instances that affect the learning performed by the next tree in the sequence. After all the trees have been constructed, the new data is predicted, and the performance of each tree is weighted according to the accuracy of the training data.



The main differences are that Gradient Boosting is a generic algorithm to find approximate solutions to the additive modeling problem, while AdaBoost can be seen as a special case with a particular loss function. Hence, Gradient Boosting is much more flexible.

Advantages	Disadvantages
• This methos has a high degree of accuracy.	• Data imbalance leads toa decrease in classification accuracy.
• It fully considers the weight of each classifier.	• Training is time consuming.

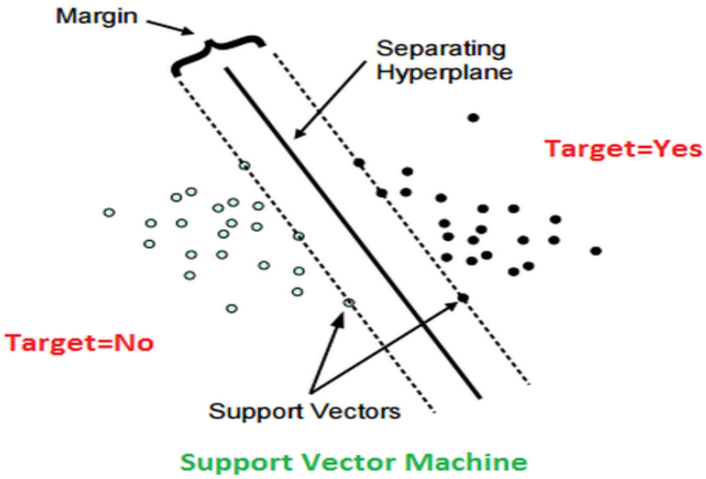
Support Vector Machine (SVM)

It is a strong classification technique that simply generates hyperplanes or lines to separate and classify the data in some feature space. To explain SVM, it is necessary to start with an explanation about Support Vector Classifier (SVC). The Support Vector Classifier is an approach for classification in the two-class setting., if the boundary between the two classes is linear. However, sometimes it is necessary to deal with non-linear class boundary; that means that the classifier will perform poorly.

Taking the above into account, the Support Vector Machine, is an extension of the Support Vector Classifier that results from enlarging the feature space in a specific way, using kernels. We might to enlarge the feature space in order to accommodate a non-linear boundary between the classes by assigning new examples to one category or another. What SVM does is that it generates hyperplanes which in simple terms are just straight lines or planes or are non-linear curves, and these lines are used to separate the data or divide the data into 2 categories or more depending on the type of classification task.

Another important concept in SVM is maximal margin classifiers. What it means is that among a set of separating hyperplanes SVM aims at finding the one which maximizes the margin M. This simply means that we want to maximize the gap or the distance between the 2 classes from the decision boundary (separating plane).

In addition, to perform linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, which is good approach when the data is not linearly separable.



Advantages	Disadvantages
- SVM works relatively well when there is clear margin of separation between classes..	- The model is not suitable for large data sets.
- SVM is effectice in high dimensional spaces.	- SVM does not perform very well when the data set has more noise (e.g. target classes are overlapping) .

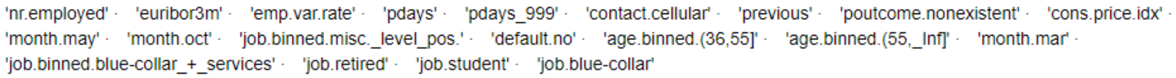
Part 2 – Benchmarking

Objective: Assess performance of algorithms and compare them utilizing different configurations in terms of resampling methods, number of iterations, evaluation metrics, variable selection and hyper-parameter tuning.

Methodology: Execution of feature selection, run a set of algorithms on a given dataset and extract performance measures from the generated information, compare them and select the best one. Then, apply hyper-parameter tuning to improve the evaluation metrics.

Feature Selection

In order to maximize the likelihood, feature selection was executed using **Fisher Score**. The list of the top 20 variables is the following:



```
'nr.employed' · 'euribor3m' · 'emp.var.rate' · 'pdays' · 'pdays_999' · 'contact.cellular' · 'previous' · 'poutcome.nonexistent' · 'cons.price.idx' ·
'month.may' · 'month.oct' · 'job.binned.misc._level_pos.' · 'default.no' · 'age.binned.(36,55]' · 'age.binned.(55,_Inf]' · 'month.mar' ·
'job.binned.blue-collar_+_services' · 'job.retired' · 'job.student' · 'job.blue-collar'
```

Setup

The benchmark experiment starts defining a list of five learners:

- Logistic Regression
- Random Forest
- Gradient Boosting (GBM)
- Adaptive Boosting (Adaboost)
- Support Vector Machine (SVM)

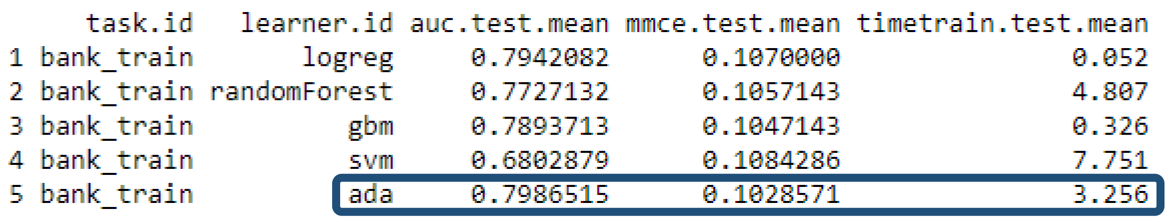
They are applied to one classification problem, in this case, bank-marketing dataset.

Secondly, benchmark was conducted 4 times. Each time, the method was configured with different resampling methods, number of iterations and evaluation metrics as follow:

Benchmark 1

- **Learners:** Logistic Regression, Random Forest, GBM, Adaboost, SVM
- **Resampling method:** Cross-Validation
- **Iterations:** 10
- **Evaluation metrics:** AUC, mmce

The results were:

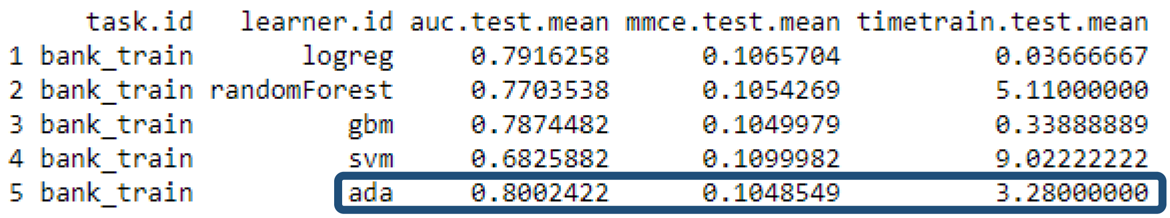


```
     task.id   learner.id auc.test.mean mmce.test.mean timetrain.test.mean
1 bank_train       logreg     0.7942082      0.1070000               0.052
2 bank_train randomForest     0.7727132      0.1057143               4.807
3 bank_train          gbm     0.7893713      0.1047143               0.326
4 bank_train          svm     0.6802879      0.1084286               7.751
5 bank_train          ada     0.7986515      0.1028571               3.256
```

It is possible to observe that the best performer is **Adaboost** with an **AUC** of **0.7986** and **mmce** of **0.1028571**

Benchmark 2

- **Learners:** Logistic Regression, Random Forest, GBM, Adaboost, SVM
- **Resampling method:** Cross-Validation
- **Iterations:** 9
- **Evaluation metrics:** AUC, mmce

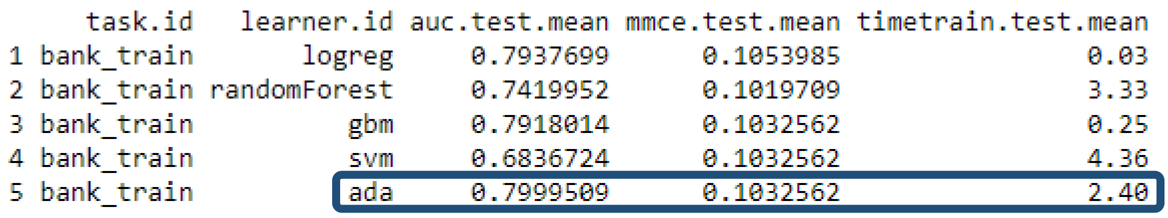


```
     task.id   learner.id auc.test.mean mmce.test.mean timetrain.test.mean
1 bank_train       logreg     0.7916258      0.1065704          0.03666667
2 bank_train randomForest     0.7703538      0.1054269          5.11000000
3 bank_train          gbm     0.7874482      0.1049979          0.33888889
4 bank_train          svm     0.6825882      0.1099982          9.02222222
5 bank_train          ada     0.8002422      0.1048549          3.28000000
```

It is possible to observe that the best performer is **Adaboost** with an **AUC** of **0.8002422** and **mmce** of **0.1048549**

Benchmark 3

- **Learners:** Logistic Regression, Random Forest, GBM, Adaboost, SVM
- **Resampling method:** Holdout
- **Iterations:** 9
- **Evaluation metrics:** AUC, mmce



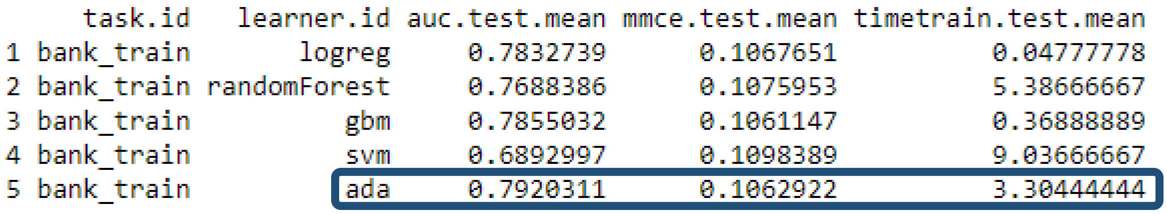
```
     task.id   learner.id auc.test.mean mmce.test.mean timetrain.test.mean
1 bank_train       logreg     0.7937699      0.1053985                0.03
2 bank_train randomForest     0.7419952      0.1019709                3.33
3 bank_train          gbm     0.7918014      0.1032562                0.25
4 bank_train          svm     0.6836724      0.1032562                4.36
5 bank_train          ada     0.7999509      0.1032562                2.40
```

It is possible to observe that the best performer is **Adaboost** with an **AUC** of **0.7999509** and **mmce** of **0.1032563**

Benchmark 4

- **Learners:** Logistic Regression, Random Forest, GBM, Adaboost, SVM
- **Resampling method:** Bootstrap

- **Iterations:** 9
- **Evaluation metrics:** AUC, mmce



```
     task.id   learner.id auc.test.mean mmce.test.mean timetrain.test.mean
1 bank_train       logreg     0.7832739      0.1067651          0.04777778
2 bank_train randomForest     0.7688386      0.1075953          5.38666667
3 bank_train          gbm     0.7855032      0.1061147          0.36888889
4 bank_train          svm     0.6892997      0.1098389          9.03666667
5 bank_train          ada     0.7920311      0.1062922          3.30444444
```

It is possible to observe that the best performer is **Adaboost** with an **AUC** of **0.7920311** and **mmce** of **0.1062922**

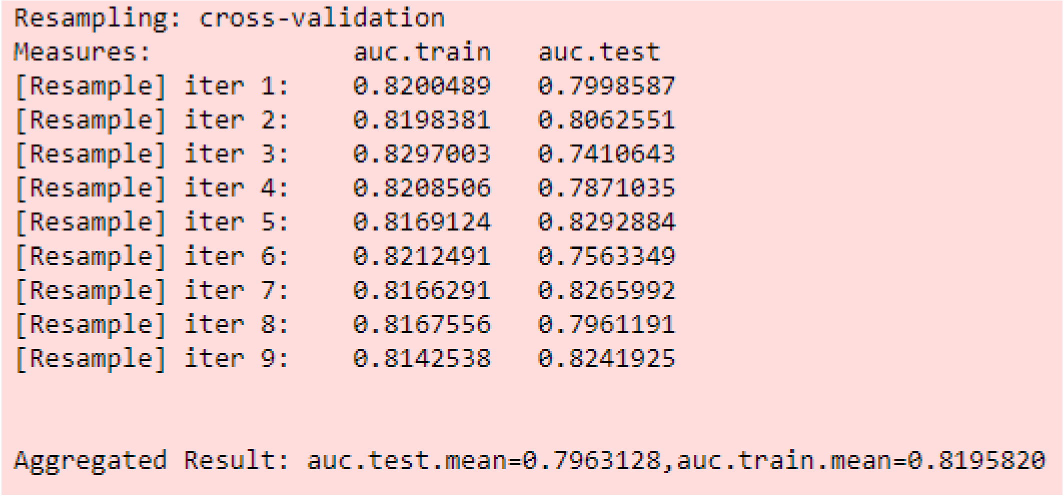
According to the previous analysis, we can conclude that the best performer is **Adaboost** and specifically, the one configured under Cross-Validation with 9 iterations. Being said this, we select this setup to continue with the benchmark analysis**.**

Given the above, the next step was the application of hyper-parameter tuning on **Adaboost** under **Cross-Validation** and **9 iterations** to train the model and to obtain the best learner.

As final steps, the model was retrained and the predictions over the target variable "subscribe" were obtained from test dataset.

Results

This table shows the results from the execution of the Adaboost algorithm.



```
Resampling: cross-validation
Measures:             auc.train   auc.test
[Resample] iter 1:    0.8200489   0.7998587
[Resample] iter 2:    0.8198381   0.8062551
[Resample] iter 3:    0.8297003   0.7410643
[Resample] iter 4:    0.8208506   0.7871035
[Resample] iter 5:    0.8169124   0.8292884
[Resample] iter 6:    0.8212491   0.7563349
[Resample] iter 7:    0.8166291   0.8265992
[Resample] iter 8:    0.8167556   0.7961191
[Resample] iter 9:    0.8142538   0.8241925


Aggregated Result: auc.test.mean=0.7963128,auc.train.mean=0.8195820
```

The final output was uploaded on Kaggle for the in-class competition.

References

Logistic Regression

https://en.wikipedia.org/wiki/Sigmoid_function

Random Forest

https://blog.tenthplanet.in/randomforest/

GBM

https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

https://docs.paperspace.com/machine-learning/wiki/gradient-boosting

Adaboosting

https://easyai.tech/en/ai-definition/adaboost/

SVM

https://en.wikipedia.org/wiki/Support-vector_machin