**Modeling assignment**

The goal of this assignment is to practice some of the data preparation techniques that were taught in session 4.

The goal of the exercise is to predict if a customer will leave our company. The dataset we are using comes from Kaggle. You can have a look to the dataset here: <https://www.kaggle.com/adammaus/predicting-churn-for-bank-customers>. The dataset itself comes with the files of this exercise.

I provide a notebook with comments in the places where you must insert the code. In this case you will provide most of the code. You can use the code from the previous exercise as a guide. In the notebook you will find comments suggesting tasks to do but you do not need to follow exactly them exactly. Use your own criteria to solve the problem.

After the data preparation steps you will fine tune the meta-parameters and evaluate the results of three different models:

* LogisticRegression: this model is quite simple and has not meta-parameters to fine tune. In this case you will use the function cross\_val\_score to measure the performance. Use the parameter scoring = ‘roc\_auc’. The function cross\_val\_score will return a vector with the score for every partition in the cross validation. What value would you choose to show the performance of this model?
* SVM (<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>): use the function GridSearchCV (grid search cross validation) to fine tune the metaparameters. In the code you will find an example of how to use the function
* GradientBoostingClassifier (<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>): gradient boosting is an ensemble method (a combitation of learners) which takes several ‘weak’ learners to obtain a strong learner. This method uses the gradients in the loss function to weight the points that are difficult to predict. Ensemble methods are very powerful and usually get very good results. Use GridSearchCV to tune the parameters ‘learning\_rate’, ‘n\_estimators’ and ‘criterion’

GridSearchCV will return a complex structure (see [https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.html#sklearn.model\_selection.GridSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html%23sklearn.model_selection.GridSearchCV)) . We are interested in the parameter values which produce the best results (best\_params\_) and the vector of best cross validated results (best\_score\_). Please, read in the documentation the meaning of these two values.

Use GridSearchCV with moderation: many parameters can take a long to return an answer. Three parameters with three values each can take about 5 minutes to execute. Be patient while the system obtain the results.

What to deliver: the notebook with the results. You must print at least the average ‘roc\_auc’ for the best version of each of the three models.