Instituto Superior de Engenharia de Lisboa
Mestrado em Engenharia Informática e de Computadores
Mestrado em Engenharia Informática e Multimédia Big data mining (MDLE)

Laboratory Class #3 — Instance manipulation
2^{nd} semester, 2023/2024 (April, 03)

Code and Report are due by April, 10

---

1. **Data Resources and Software Tools.**
   For this laboratory class, we will consider a subset of the work assignment' dataset.

   Table 1: Dataset with d features, c classes, n instances and the corresponding problem/task to solve.

Dataset	d	c	n	**Problem/Task**
Influenza	545	2	2190	Classification using an imbalance dataset.

   Regarding software tools and code, we will consider:

   - R, https://cran.r-project.org/;
   - R Studio, https://www.rstudio.com/;
   - R packages `dplyr`, `Sparklyr`, `data.table`, `caret`, `e1071` and `smotefamily`;
   - The code handed with this guide.

   As a pre-processing step, you may need to: *(i)* install these tools; *(ii)* check if SPARK is correctly installed . *(iii)* analyse the R code; *(iv)* Run the code inse the regions `Preparation`, `Spark setup` and `Load data`

---

2. **Visualise the dataset**.

   (a) Use the sparklyr `sdf_schema` function to check the schema of the `df` variable.

   (b) Check the content of the SPARK data frame *df*, using the `head` function.

   (c) Use the `stopifnot` function to guarantee that the number of columns and rows in *df* is correct. To achieve this goal, apply the `nrow` and `ncol` functions (or the equivalent in Sparklyr), and compare the values with the ones in Table 1.

3. **Feature Selection**.

   (a) Use the magrittr's pipe operator, %>% and the `select` function to reduce *df* features to the features in the indexes 1, 2, 5, 6, 9, 10, 11, 14, 16, 17, 19, 21, 24, 25, 26, 31, 32, 33, 34, 35, 41, 44, 49, 50, 54. Store the resulting SPARK Dataframe in the `df.sel` variable. Notice that the first feature is the dependent variable, named `CLASS`.

   (b) Use the `head` function to overview the resulting dataset.

4. **Use of generic sampling techniques**.

a) Apply the sparklyr `sdf_random_split` function to produce two datasets: one for training (2/3) and other for testing (1/3). Use the seed value 123, for this and all random functions from this point forward.

b) Use the R `table` function to determine the number of instances for each class in both datasets. Explain why this function cannot be used directly on `df.train` and `df.test`.

c) Use the `ml_random_forest` function to generate a classification model, with the formula: "`CLASS ~ .`".

d) Using the helper function `mdle.printConfusionMatrix` and `ml_predict`, check the performance of the model. Consider it as the *baseline model*.

5. **Using imbalanced correcting sampling techniques**.

(a) Using the training set from 4.a), apply an undersampling technique to balance the number of cases of each class. Use the function `sdf_sample`. To calculate the fraction use the functions `nrow` and `collect` (or `sdf_nrow` alone) on the `df.pos.train` and `df.neg.train` variables, and combine them with `sdf_bind_rows`. What is the number of instances for each class in the training set after the undersampling?

(b) Repeat points 4.c) and 4.d), and compare the results with the previous models. Are they better? Are they worst?

(c) Using the training set from 4.a), apply an oversampling technique to balance the number of cases of each class. What is the number of instances for each class in the training set after the oversampling?

(d) Repeat points 4.c) and 4.d), and compare the results with the previous models.

(e) Apply *Borderline-SMOTE Sampling* to balance the number of cases of each class, using the `BLSMOTE` function from the `smotefamily` package. The first parameter is a data set without the class. During the oversampling process, use only `R data.frame` variables. Indicate what are the values that you used for K, C, and `method` parameters.

(f) Repeat points 4.c) and 4.d), and compare the results with the previous models.

6. **Comparision**

(a) Based on the results achieved, what sampling technique do you think is probably better for this dataset, considering the problem in Table 1? Present a table, where the best results are highlighted in bold. Explain. Consider the example below:

	False Positive Rate	**Accuracy**	**Kappa**	**Pos Pred Value**	**Neg Pred Value**
Baseline	**0.xx**	0.xx	0.xx	0.xx	0.xx
...	...	...		...	...
Borderline smote k7	0.xx	**0.xx**	**0.xx**	**0.xx**	**0.xx**