Technical Document – Datathon
GROUP C

Our technical documentation consists on two Jupyter notebooks in which we have developed the data inspection, preparation and modelling.

1. Final_code.ipynb: In this document you will find the data inspection and preparation of our training and test dataset, that will be used for the modelling.
2. ML_pipeline.ipynb: This document contains the pipeline of our machine learning model, using the training dataset generated in the previous notebook, and the final predictions for 2019.

The overall flow of the execution will be:
Execute all code of Final_code.ipynb, until line of code [49].
Execute all code in ML_pipeline.ipynb.
Execute the remaining code on Final_code.ipynb, from [52] onwards.

The codes need to be executed from the same directory as the following files that are needed as an input:

CDS_2016_va.csv
CDS_2017_va.csv
CDS_2018_va.csv
CDS_2019_NO_LABEL.csv
Maestros.csv (provided)

The rest of the provided files will be generated as a result of the execution.

Final_code.ipynb

Firstly, we approach the general aspects of the model. For training we will use all the CDS datasets, except the 2019 file, along with some information from the provided Maestros.csv file. The test set will consist on the 2019 file, and it will have a Boolean prediction that determines if that specific teacher will or will not change into “non-use” of editorial material.

Training dataset:

When looking at the data, you realize that each client Id corresponds to a school, but this ID is obviously repeated and the rest of the variables do not match. In order to have unique rows for each of the teachers, we create an **artificial id** combining: client id, Year, the subject, the type of material, language in which is taught and type of support. This id is created for the 4 CDS files, as it will be what we base our predictions for 2019, each artificial id will have a Boolean (1/0) Target, where 1 = changing from one year to the next one, to Non-use.

The predictions will be based on the Editorial Group of the previous year. If it changes from one editorial group, to 90, that is a change to "non-use", which is what we want to predict at the end. Because of this, we create data frames for each year to analyze this change in *Grupo Editorial* calling it target. For each data frame, we will create another variable **target_variable** that will have the switching of the editorial group. We make 2 target data frames: **target2017** with the changes from 2016 to 2017, and **target2018** for the changes from 2017 to 2018. If it changed from another editorial group to 90, is a 1, and if it stayed the same or changed into something else than 90, is a 0. We then merged the two target data frames to their original CDS files and got final_target2017, and final_target2018 with the following information:

	art_id	target	Id_Cliente	Ano	Curso	Asignatura	TipoMaterial	GrupoEditorial	Lengua	TipoSoporte	Clase	ValorClase
545688	21066335821131	0	210663	2018	35	82	1	1	13	1	10.0	282.8

These two data frames contain then information from 2016, 2017 and 2018. That is why we do an outer join based on the all the variables except the editorial group, the target, class and value class. We end up with information from the 2017 target file as the x variables and from 2018 with the y variables, along side shared information. To have a consistent dataset that actually contains all the information accurately, we map columns according to the year that each record belongs to and save information about the actual editorial group and the previous one, as this is the information we will have in our test set. At this point, our training set has the following variables:

art_id	Id_Cliente	Curso	Asignatura	TipoMaterial	GrupoEditorialPrevio	Lengua	TipoSoporte	Target	Ano	GrupoEditorialActual	Clase	ValorClase
21066335821131	210663	35	82	1	90.0	13	1	0.0	2018.0	1.0	10.0	282.8

Finally, to have more information about each teacher and each school, we merge the previous dataset to the relevant information from the **Maestros** file: Comunidad autónoma, location as in longitude and latitude and the *titularidad*, if the school is private, public or catholic, as we think this information may easily influence on the decision taken by each teacher. To only have one column regarding the editorial group, we apply a lambda function that compares both grupo editorial variables and changes the target to 0 for the rows that do not exist in 2018 and the previous editorial group is 90 and also for the rows that turned to 90 in 2017 and stayed as 90 in 2018. We code the final added information and we have our final training set: **final_training.csv**

Test dataset:

For the creation of our test set, we will add the previous *Grupo Editorial* (2018) to the 2019 dataset, so that we can use that variable to predict the target as well. We then add the *Maestros* file information we also added in the training set, and we have our final test set: **test.csv**

Our two datasets will have the following variables, only differing in the presence of the Target variable in the training, which will be our actual target variable in 2019:

final_training.csv → Id_Cliente, Ano, Curso, Asignatura, TipoMaterial, Lengua, TipoSoporte, Clase, ValorClase, art_id, Grupo Editorial Previo, Latitud, Longitud, ComunidadAutonoma, Titularidad and Target
test.csv → Id_Cliente, Ano, Curso, Asignatura, TipoMaterial, Lengua, TipoSoporte, Clase, ValorClase, art_id, Grupo Editorial Previo, Latitud, Longitud, ComunidadAutonoma and Titularidad

All the coding done until this point will generate the required files to execute the Machine Learning Pipeline code. This pipeline will generate the predictions.csv required for finally coming up with the business approach. This code will be explained below, important to mention that is done in another notebook only focused on the model.

ML_pipeline.ipynb

1. Reading and preparation

In this notebook we will do the best possible modelling based on several aspects of Machine Learning. We read our training and test set, and decide to drop some columns based on the following reason:

- Unnamed: 0 from training and test. This is an auto-generated id we do not need
- Ano from training and test: the test *Ano* variable can only have one value (2019) and there are two possible values for the training.
- Longitud and Latitud from training and test: as the client ids are created for each school, the longitude and latitude will be the same for each id_Cliente, being directly correlated.
- GrupoEditorialActual from training: we will not have this column in the test set, so we cannot train the model with this variable.

2. Data cleaning

Change our variables in both training and test into the dtypes that they are, basically change into **category** the majority of columns.

3. Balancing and splitting our data

Our model will consist on a classification model, that will either predict a 1 or a 0. Our observations only consist on around 8% of observed ones, meaning that the number of observations for the 0 class is significantly larger that the 1 class. This is a problem because ML classifiers fail to cope with imbalanced training datasets because they are sensitive to the proportions of different classes, and as a consequence, the algorithm tend to favor the majority class, and this may lead to misleading accuracies. Given that what we are focusing on is the predictions of the minority class, we can find high accuracy without balancing our data because this is the product of the correct classification of the majority class. Therefore, we decide to balance our data to have the same number of observations for each class, and we will go forward with the splitting and training of our model with this balanced training set: **bal_training**

We then split our training set into X and y, and this into the train and test: **X_train, X_test, y_train, y_test.** The training of our model will be done with this for datasets.

4. Baseline Model

For each of the models we have, you will find in the notebook the confusion matrix and a classification report with the precision, recall, f1-score and support.
The results provided are based on the balanced dataset, before balancing it seemed that we had a better accuracy, 0.96, but we realized that the sensitivity of the model was very bad and then we moved to our actual balanced training.

We run a baseline model with the actual information before going forward with the feature engineering.

- Accuracy of 0.6037 for logistic regression: this will find a line to split the data into the two classes

We run a random forest classifier model as we know that the decision boundaries adapt better to the data and it shows more flexibility.

- Accuracy of 0.8365 for random forest.

Having these two baseline models, we go forward with the deeper analysis

5. Outliers

We look for outliers in the only two numerical columns we have, *Clase* and *ValorClase* and delete them for a more accurate model, and re-run the random forest baseline model without the outliers

- Accuracy of 0.8077

6. Feature engineering

We create a new feature based on the numerical variables mentioned before, as one depends on the other one, the correlation between them is very high and we want to avoid multicollinearity. Our new feature will be a ratio between the value of the class and the class itself, and its meaning is the monetary value of each student in the class. We re-run the model with this new feature:

- Accuracy of 0.8183

7. Cross Validation

We do a cross validation process with 5 folds to find the optimal and final model:

- Accuracy: 0.82
- F1-score(1): 0.83
- F1-score(0): 0.80

8. Running the Final model over all the Training Data and get the predictions for our test set: **predictions.csv**

At this point, we have to come back to our previous notebook (final_code), in which we will merge the predictions with the 2019 CDS file. This new file will be the one used to perform Business Analysis and generate insights. It contains geographic and client features: **final_2019.csv**.
Additionally, we generate the final delivery file with the required format to assess the accuracy of the obtained predictions: **CDS_2019_PRED.csv**.