Nicolás Vila - 240230
Laia Marinel·lo - 240256

Universitat Pompeu Fabra Barcelona

# Machine Learning: Supervised Learning Project

## Data Gathering:

**[EX1] [REPORT]:** In this project we will be working with a dataset about IoT companies. It is fundamental to understand and gather information about the dataset before starting applying different algorithms. With the help of the function of Matlab **info()** we note that our data has a total of 13335 rows and 10 columns. We noticed there are 3 data types: object, int and float. The variable type of "City" is "object" because it is formed by a string. Moreover, the variable with more nulls is by far the variable "Revenue" with a total of 4746 nulls while City, Customer_Flag, CNT_EMPLOYEE, Mobile_potential had none of them.

**[EX3] [REPORT]:** Now, we will follow with the next part of understanding our data which is plotting the main statistics of each variable in a box plot differentiating when the label of the company has a customer label or not. When doing that, we notice that the *custumer_dt* boxplots have a higher interquartile range (Q3-Q1) than the ones in *noncustomer_dt*. That is because the data belonging to customer_dt is in general more dispersed. We also noticed that in general, the values for the variables in *customer_dt* are higher in average than the values in *noncustomer_dt.* To see which mean value was higher for this variable, we used the describe() function and we checked, as we saw visually, that the dataset customer_dt has a higher mean for the variable CNT_EMPLOYEE than *noncustomer_dt.* Furthermore, we can clearly observe that generally, the variables in the *noncustomer_dt* dataset have more outliers than *customer_dt.* More specifically, the variable 'Revenue' had more outliers in noncustomer_dt than in customer_dt, which can be easily seen in the boxplot.
Finally, we printed the different quantiles Q1, median (Q2) and Q3 for Revenues and Mobile_potential. We can see each values in the following image:

```
Q1 for 'Revenue' in customer_dt is: 1047500.0
Q2 (median) for 'Revenue' in customer_dt is: 2200000.0
Q3 for 'Revenue' in customer_dt is: 4195000.0


Q1 for 'Mobile_potential' in customer_dt is: 2090.6967281537
Q2 (median) for 'Mobile_potential' in customer_dt is: 2401.464692530968
Q3 for 'Mobile_potential' in customer_dt is: 2826.2351826061667


Q1 for 'Revenue' in noncustomer_dt is: 902986.0
Q2 (median) for 'Revenue' in noncustomer_dt is: 1750000.0
Q3 for 'Revenue' in noncustomer_dt is: 3501123.5


Q1 for 'Mobile_potential' in noncustomer_dt is: 1975.5165190653966
Q2 (median) for 'Mobile_potential' in noncustomer_dt is: 2277.9727974861535
Q3 for 'Mobile_potential' in noncustomer_dt is: 2631.926166103982
```

**[EX5] [REPORT]:** After calculating the ratio values for the 'City' variable in each dataset we saw that there were way more different cities in noncustomer_dt than in customer_dt. This could be explained by the fact that the size of noncustomer_dt is also way bigger than the size of customer_dt. Therefore, it also makes sense that the city with highest ratio in noncustomer_dt was 'Köln' (0.016) while the one with highest ratio in the dataset for customers was 'München' (0.024). This could also point out that there is a meaning behind these ratios -even though the ratio is not enough evidence to be sure about it– and that it might be more likely that a company in Munich becomes a customer than in any other city.

**[EX6] [REPORT]:** The length of X_train and X_test datasets are 4692 and 1173 respectively. Knowing that the final dataset's length is 5865, if we set the test_size as a 20% of the final_dataset, the length of X_test should be 0.2*5865 = 1173. Therefore, it is obvious that the length of our datasets are aligned with what we have set in the split.

**[EX7] [REPORT]:** We can see clearly in the histogram that the dataset is imbalanced as there are way more rows with a label '0' than with label '1'. This could negatively affect our model because it might be biased towards a certain label implying that our classifier could be wrongly trained, and could still give us high accuracy.

Model training:

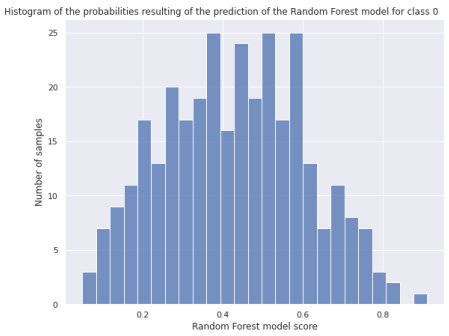
**[EX12][REPORT]:** After analyzing the printed data, we can see that the model with highest accuracy is the Decision Tree with a 0.57 value. Overall, the Decision Tree model gives a better and more balanced recall, precision and accuracy than SVM. Therefore, the one we would recommend to classify both classes is the Decision Tree model.

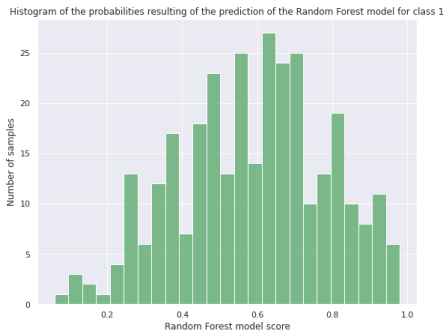
**[EX13][REPORT]:** We wouldn't say that a VotingEnsemble model gives better results than the previous algorithms. We can observe that it is a better classifier for class '0' as the recall of this target is considerably higher than the other models. However, accuracy is slightly lower and recall for the class '1' is the lowest of all of them. For instance, the best classifier for customer_dt would be the Decision Tree since the recall for target '1' is the highest of them all. However, the best classifier for target '0' could be debatable between the voting ensemble method, with the highest recall, or the decision tree, which has higher precision for target '0'.

**[EX14][REPORT]:** For this model, we observe several differences in comparison to the ones previously seen. In the first place, it is the model with the highest accuracy. It is also the model with the highest recall for class '1' which means that it is a great classifier for customers and it also has a high precision for this same target. Regarding its performance in class '2', we note that the precision is the highest between all the models we've seen so far. The recall is 0.65 which is higher than the one seen in the Decision Tree model which was, until now, probably the best model.

Therefore, because of all of the things stated above, we believe that this model has an overall better performance than the ones we've seen until now along with the Decision Tree which is also not far from this model's metrics, but indeed, are a little lower.

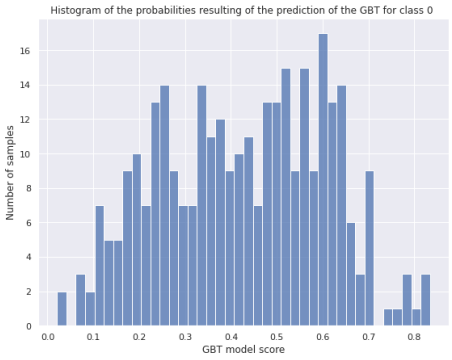
**[EX15][REPORT]:**

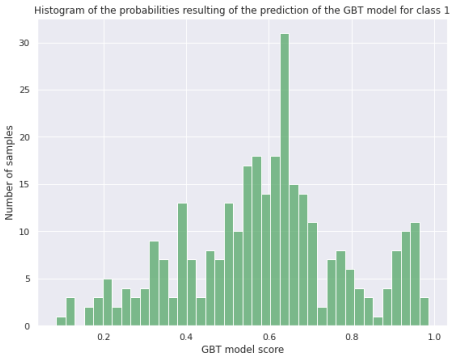



**[EX16][REPORT]:** After taking a look at the obtained values for precision, recall, and confusion matrix we can say that the model (Gradient Boosting) would probably be the best seen so far. The performance regarding customer classification was quite balanced and had high values. More specifically, it has almost the same precision as the Random Forest Classifier and the highest recall between all the models seen so far. In general terms, it has really good balanced metrics. On the other hand, when we checked the metrics for target '0', we noticed that it didn't have the best recall value (0.61) but it had the best precision value (0.68) which balances its performance. In general terms it was the best classifier seen so far for customer dataset. Moreover, it was on the level of Random Forest regarding classification for class '0'. Finally, it also had the highest accuracy value seen so far. We believe that this model had the better and most balanced metrics for this classification task.

**[EX17][REPORT]:**




After analyzing all 4 histograms, we believe that the model classifying better is the Gradient Tree Boosting. If we take a sharper look at the Gradient Tree Boosting histogram, we see that for class '1' (green) there are way less number samples with low probability of being one than in Random Forest's histogram.

The samples are mostly in the right side of the Histogram which means that they have a high probability of being 1. Regarding the '0' samples (blue), we have observed a different behavior: few of them have a very high probability of being 1 but the vast majority are on the left side of the graph, which means that they have a low probability of being 1. This phenomenon is similarly replicated in Random Forest. However, in general terms, the model that classifies the best both classes is the Gradient Tree Boosting one. That is why we opted for this model and it would be the one to use in order to send noncustomers to the marketing manager.

**[EX18][REPORT]:** In order to choose the cut-off , we will test different values and then choose the one that gives us the higher precision and recall for the Gradient Tree Boost algorithm. We believe that the best and more balanced values for recall and precision were found with a cut-off of 0.5 and 0.55. Moreover, with these values for the cut-off the accuracy was 0.66 in both, which was the highest between all cut-offs that we tried. In order to know the number of non customers that we will send to our sales manager, we need to check the second element of the first row of the confusion matrix, which tells us the number of false positives in our classification. That is, the number of non customers classified as customers. Therefore, we see that we will send around 124 companies, if we choose a cut-off of 0.5 and 95 if we choose 0.55. Both of them would be valid given that they have high and balanced precision and recall values as well as high accuracy.

**[EX19][REPORT]:** After executing and ordering the features by importance, we printed each of them along with their respective importances. We got the following output:

```
Top: 0 , feature importance:  ['Mobile_potential', 0.2300893370343828]
Top: 1 , feature importance:  ['Revenue', 0.1377348004767908]
Top: 2 , feature importance:  ['Legal_Form_Code', 0.12131466068247054]
```

**[EX20][REPORT]:**

- **In this case, which is the target variable? Which are target = 0 samples? And target = 1?** In this case the target variable would be all those customers that have already acquired the new IoT tariff or not . That is, Target = 0 samples would be the ones not having the new IoT tariff hired and Target = 1 samples the ones having it hired.

- **Would you add these 3 new variables to the dataset?** As stated, adding more variables would imply a higher computational cost. We would need to study whether these variables would be significant or influential in our classification task. In the case where it introduces a better classification and we are willing to pay the computational cost, then yes, we could add these variables knowing that there's more information we can gather from our customers than from non-customers. Adding these variables can give us, on average, information related to what common features users have of this tariff with respect to these 3 parameters.

Nicolás Vila - 240230
Laia Marinel·lo - 240256

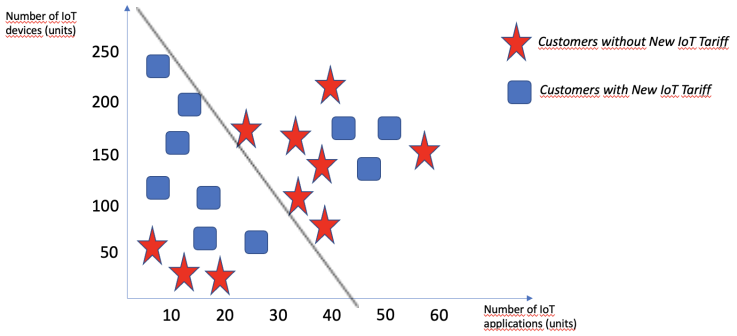
Universitat Pompeu Fabra Barcelona

● **Will the training dataset be balanced or unbalanced?** The training dataset will be unbalanced as the number of samples of customers without Mobile Tariff will be way lower than the number of customers with a Mobile Tariff hired given its popularity.

● **Describe in terms of Number of IoT devices (units) and Number of IoT applications (units) the pattern of target 1 customers**

We observe that the pattern for target 1 customers is that they tend to have a lower number of IoT applications. Moreover, the fewer IoT applications they have, the higher the number of IoT devices the customer will have.

**●Draw a plane to separate both classes**




**●Is the training dataset balanced? Justify your answer.**

It is indeed balanced as we have 10 samples of each class.

**●According to the previous plane, which are the customers to be phoned to sell the New IoT tariff?**

We see that the customers to be phoned would be those who have been classified as customers with the new IoT Tariff but nevertheless, they are not (false positives). This means that they would fit in the class of the new Tariff owners and they are more likely to acquire the new Tariff.

**●Could you estimate the precision and recall of the classification?**

For the clients that do not have the New IoT Tariff, we observe that the precision is 7/10 = 0.7, and the recall would be the same because from all of those rows with target '0'- 10 in total -, 7 of them have been correctly classified. On the other hand, for customers that have acquired the new tariff, we observe that the precision and recall is also 0.7. Overall, we have good metrics for the classification of both classes and therefore, the line we drew would be a good classificator.

**We hereby declare that, except for the code provided by the course instructors, all of our code, report, and figures were produced by ourselves.**