Regarding **data preprocessing**, we analyzed whether our dataset had missing values and fortunately, it didn't. Then, we checked the distribution of instances and our dataset was slightly imbalanced as we have seen before. Then, for the numerical variables we applied the *StandardScaler*, and for categorical we enforce the *OneHotEncoder* dropping the first category to avoid multicollinearity.

Models - Classification Problem

Regarding the models, we follow the steps we covered in class. For simplification purposes, we will only analyze the *accuracy* and *recall* scores since they are the most important from our perspective.

Regarding the **KNN** model, we start by applying the *KNeighborsClassifier* with a k=3. Then, we tried several values for k. We also apply a regularization technique by setting the *weights='distance'* and also change the distance metric used to "Manhattan". Lastly, we also applied the *RandomOverSampling* technique to address the little imbalance but the best results were with the normal model with a k=3.

With respect to the **Random Forest** model, we applied the *RandomForestClassifier*. Then, as the results were poor, we establish the parameters: *ccp_alpha*, *class_weights*='balanced', and *n_estimators*. Lastly, we applied a GridSearchCV on the most common parameters applied to this model, and the best results were achieved here.

We decided to go further and analyze two different models with a lot of potentials to improve our results.

Regarding the **Support Vector Machine (SVM)** and the **XGBoost** models we first apply the normal conditions using SVM and XGBClassifier, respectively, which leads to the presented results. In order to improve them, we apply optimization in each model hyperparameters: C, *kernel*, and *gamma* in the SVM and *alpha* and *lambda* in the XGB.

We can conclude that the four previous models are not good at explaining employee satisfaction problem, which means that they are not able to make accurate predictions.

With regard to the **Trees**, we follow the same procedure as random forest and once again the best results were achieved throughout the *GridSearchCV* regarding the most common hyperparameters used in trees.

Regarding the **Neural Network** model, we first apply *MLPClassifier* with 3 hidden layers with 3, 4, and 5 neurons. Then, we apply the *L2 regularization* and look for the best *alpha.* Lastly, we apply a *RandomOverSampling* technique. The best results were achieved with the first attempt.

Also, in both of these models, we found results quite strange, with better performance in the test set than in training, which may be due to the excessive optimization we did that lead to the underfitting of the data.

Concerning the **Naive Bayes**, we start by applying the *Gaussian NB.* The results were not so different but we still did a *GridSearchCV* on var*_smoothing* and we apply the *RandomOverSampling* technique.

When estimating the model through ***LogisticRegression***, we impose the *class_weight='balanced'* and *max_iter*. The results were relatively close but still, we used the *Lasso Regularization.* Finally, instead of imposing a value on *C*, we created a hyper by assigning it several values and did a *GridSearchCV*.

Despite presenting values that are not very high, in the context of the problem of employee satisfaction, we can consider these two models more or less good because they can predict, on average, almost half of the positive instances correctly and they are also able to apply the knowledge learned during training to make accurate predictions in unseen data. In this sense, we can say that theses models are the most stable and robust among all 8 we have estimated.