**MODELLING PLANS FOR STAGE 2**

**DATA SCIENCE AND DECISION MAKING**

**MSc. INTELLIGENT SYSTEMS AND ROBOTICS**

**JUAN DAVID VELASQUEZ RESTREPO – ID: 2400901**

**MODELS TO USE:** The expected models to use are the following:

* **SVM:** This model can handle eitherlinear or non-linear data by offering a variety of combinations not only for but the kernels, but for the rest of hyperparameters that might be tested and analysed in the data, offering a wide range of possibilities where each can impact the performance of the model. Also, the SVM is generally considered as a model that can be resistant to overfitting (given that the classes are highly imbalanced) and also, does not require a high computational resource.
* **Logistic Regression:** This one might be considered as the simplest model to be tested, mainly because this one offers a high interpretability to understanding what coefficients impact on the relationship between features and also, it is an efficient and fast training model.
* **KNN:** This model is chosen becausecan adapt to non-lineal features and, does not make assumptions about the data distribution. In addition, KNN can handle multiclass labels without any additional modifications and lastly, does not require training, making this model quick to be understood and implemented.
* **Random Forest:** Since this classifier stacks several simple classifiers, this model is highly resistant to overfitting and performs better in generalization for imbalanced classes. Also, can identify what features are the most relevant to determine the label and offers versatility to handle lineal and non-lineal data.
* **XGBoost:** This classifier is commonly used for handling imbalanced classes,can deal with lineal and non-lineal data and can be highly interpretable since it allows the code to represent the features importance through metrics such as gain, cover and weight.

**EVALUATION METRICS:** Since the class distribution is not balanced, accuracy is not suitable for this project, instead, these will be the evaluation metrics to be used:

* MCC (Matthews Correlation Coefficient)
* F1 Score
* ROC curve and AUC

**MODELLING PLANS:** Once the complete dataset is scaled, reduced in features, and skew corrected, the next step is to build a pipeline for each model, where each will follow these steps:

* **Inner Cross-Validation:** A cross-validation of 5 folds will be performed using the training set to optimize hyperparameters and select the best configuration for the model.
* **Outer Cross-Validation:** The optimized model will then be evaluated using another 5-fold cross-validation on the entire training dataset, ensuring a robust estimation of its generalization performance.
* **Hyperparameter Tuning:** The inner cross-validation will be used to test different hyperparameter combinations depending on the model, selecting the ones that maximize the performance for the MCC.
* **Final Model Selection:** After completing the nested cross-validation process, the model with the highest value of MCC will be selected for final testing.
* **Performance Evaluation:** The selected model will be assessed on the test set using various metrics such as MCC, F1-score, and ROC curve and AUC.

**ALTERNATIVES IN CASE OF LOW PERFORMANCE:**

If the model's performance remains low (MCC < 60%) even after the performance evaluation phase, the following alternatives can be applied to improve results in both the training and testing stages:

* Increase the variance retained by PCA (e.g., setting n\_components=0.99).
* Reduce the range of columns removed due to low correlation with the target variable.
* Reduce the range of columns removed due to high correlation between the features.
* Ensure a stratified train-test split to maintain balanced class distributions across both sets.