

# Informe TP 1 Checkpoint 3

- **Clasificación con KNN**

Para la clasificación de KNN usamos el 25 por ciento de los datos en test y el 75 restante en entrenamiento y usamos un `random_state = 4`. En la búsqueda de hiperparametros, utilizamos el Random Search y los fuimos ubicando de manera aleatoria de tal forma que nos de la mejor predicción. La mejor predicción que tuvimos fue con un k-folds de 5, con 5 iteraciones y considerando un rango de 1 a 50 para el número de vecinos y un `random_state = 2`.

- **Clasificación con RF**

En la clasificación Random Forest empezamos armando uno con los hiperparametros random. Luego de esto buscamos los mejores hiperparametros con GridSearch CrossValidation, utilizamos 5 folds para esto, basándonos principalmente en el que nos dio una mejor precisión media. De esta forma armamos el mejor tipo de RF.

- **Clasificación con SVM**

Para el SVM fuimos comparando cómo mejora la precisión y el recall al trabajar con datos escalados contra datos no escalados. Y dentro de los datos escalados, como cambia la matriz de confusión entre datos escalados con Min Max y datos escalados con normalización. También se comparó con PCA. Y finalmente variamos el kernel entre 3 distintos: lineal, polinómico y radial. Como pudimos concluir, el lineal fue el que mejores resultados nos dio ya que fue el que mejor pudimos entrenar. En cambio el polinómico, tuvimos que limitar la iteración de su entrenamiento ya que este no terminaba nunca.

- **Clasificación con Ensamble Híbrido Tipo Stacking**

En la clasificación con Ensamble Híbrido Tipo Stacking utilizamos 3 modelos base siendo estos: RF, SVM y KNN. Luego para el meta modelo utilizamos regresión logística.

- **Clasificación con XGBoost**

En la clasificación con XGBoost buscamos acotar los hiperparametros que consideramos más significativos mediante cross validation y luego los optimizamos RandomizedSearchCV.

- **Clasificación con Ensamble Híbrido Tipo Voting**

En la clasificación con Ensamble Híbrido Tipo Voting no buscamos optimizar parámetros más allá de los que ya encontramos previamente. Hicimos una pequeña validación con 5 folds.