FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 1 - Checkpoint 2

Grupo 21 - 'df' is not defined

[7506/9558] Organizacion de Datos
Primer cuatrimestre de 2023

Alumno	Padron	Email
Martin Pata Fraile de Manterola	106226	mpata@fi.uba.ar
Tobias Emilio Serpa	108266	tserpa@fi.uba.ar
Juan Francisco Cuevas	107963	jcuevas@fi.uba.ar

1. RESULTADOS OBTENIDOS:

Antes de comenzar a probar los modelos, decidimos normalizar los datos con un escalador MinMax en el preprocesamiento de datos. esto se debe a que en SVM se necesita tener los datos normalizados para correr los modelos.

En primer lugar, comenzamos probando el algoritmo K-Nearest Neighbors, con sus 3 variaciones posibles (Kd_Tree, Ball_Tree y Brute). El que mejor resultado arrojó fue Brute con el estimador de distancia Canberra.

Luego continuamos con el Random Forest, donde de manera experimental fuimos buscando los mejores hiperparametros y a partir de eso armamos nuestro modelo final con los mejores valores arrojados por el gridsearch. Este modelo ya arrojó resultados superiores a KNN.

Para las Support Vector Machines se probaron los 3 tipos de Kernel: Lineal, Polinomico y Radial, obteniendose el mejor resultado de este ultimo. Consideramos que se podían haber encontrado mejores hiperparametros para generear mejores modelos, pero nosa vimos limitados a continuar con la busqueda debido a el gran consumo computacional del algoritmo SVM.

El siguiente tipo de modelo trabajado fue el XGBoost, donde, en primer lugar elaboramos un amplio gridsearch con una cantidad considerable de variables y sin normalizar los datos, obteniendo asi los mejores hiperparametros para esa prueba. Luego realizamos un grid search con una cantidad de variables mas acotada (las que considerabamos que iban a mantener los resultados, pero reduciendo los tiempos), y con una normalización MinMax de los datos. Con esto logramos reducir los tiempos de ejecucion y mejores resultados.

Para ir concluyendo, probamos los modelos de ensamblado. Comenzamos con un voting de modelos genericos, para luego obtener un mejor resultado generando un voting con los mejores hiperparametros encontrados anteriormeten para cada modelo. Luego utilizamos el stacking con la misma modalidad, iniciando con modelo genericos para luego probar nuestros modelos propios (en ambos casos se obtuvieron mejores resultados al utilizar nuestros hiperparametros previos).

2. MEJOR MODELO:

Nuestro mejor modelo de prediccion sale del XGBoost, en el caso en que usamos un grid search mas acotado, pero con datos normalizados. Trabajamos de manera experimental y con los datos del XGBoost anterior, para encontrar los mejroes hiperparametros que nos permitó conseguir nuestro mejor modelo.