

Trabajo Práctico 1

Reservas de Hotel

Integrantes

Apellido	Nombre	Grupo
Lopez	Francisco	17
Corn	Franco	17
Queirolo Dominguez	Cristian Daniel	17

Informe

Introducción

En esta parte del trabajo práctico realizaremos el checkpoint 3, buscando ensambles y luego tomaremos los mejores 3 para el uso de stacking y voting. Usando stacking y voting veremos cuánto mejora nuestra performance.

Objetivos

Checkpoint 3: Ensamblés

- a. Construir un clasificador KNN optimizar sus hiper parámetros mediante k-fold Cross Validation
- b. Construir un clasificador SVM variando el kernel y los parámetros.
- c. Construir un clasificador RF y optimizar sus hiper parámetros mediante k-fold Cross Validation
- d. Construir un clasificador XGBoost y optimizar sus hiperparámetros.
- e. Construir un ensamble híbrido tipo Voting y otro tipo Stacking.
- f. Evaluar la performance de todos los modelos en entrenamiento y validación, explicar todas las métricas y mostrar la matriz de confusión.
- g. Generar predicciones con el set de test y realizar los submits correspondientes en la competencia de Kaggle.

Desarrollo

Knn y Svm

En knn y svm buscamos crear los ensambles y optimizar sus hiper parámetros. Los dos ensambles resultan bastante “regulares”, muy pesados en compilación y en búsqueda de hiper parámetros (knn un poco menos que svm). Los resultados de las

métricas fueron aceptables con una media de 0.80 con hiper parámetros, los ensambles no resultan tan buenos como predecimos antes de iniciar. Siendo svm el más lento y peor de los dos. Asumimos que también, debe tener que ver con la búsqueda de hiper parámetros, que tardó demasiado y además no nos mejoró demasiado nuestras métricas.

Xgboost y Random forest

En Xg boost y Random forest, cambia bastante todo. Rápidos en ejecución y su búsqueda de hiper parámetros no fue tan ardua como en knn y svm, ambos con métricas muy buenas 0.88 de promedio (ambas) con hiper parámetros. Estos dos ensambles fueron muy superiores a knn y svm.

Como vimos en la práctica, estos dos ensambles son mejores que knn y svc y su performance fue muy superior.

Voting y stacking

Para stacking y voting elegimos nuestros mejores tres ensambles, vemos que al utilizar voting obtenemos un f1_score muy parecido al que obtuvimos con xgboost y en stacking subimos un poco nuestro f1_score . Esto se debe a que nuestros modelos más poderosos (xg y random forest) predicen muy parecido entre sí.