

Informe del Trabajo Practico 1

Facultad de Ingeniería de la Universidad de Buenos Aires



Grupo 09

Castro Martinez, Jose Ignacio
Padrón: 106957
email: jacastrom@fi.uba.ar

Douce, German Alejandro
Padrón: 106001
email: gdouce@fi.uba.ar

Orsi, Tomas Fabrizio
Padrón: 109735
email: torsi@fi.uba.ar

KNN

El modelo Knn es capaz de predecir con una precisión aceptable. Por un lado el modelo base tiene una precisión aproximada de 0.70, al someterlo a la búsqueda de hiperparametros y observar su comportamiento mientras se agregan más vecinos observamos que puede llegar a mejorar, de tal manera que toma una precisión aproximada de 0.74, lo cual representa una mejora considera con relación al modelo base.

El modelo no representa una mejora predictiva con relación al modelo anteriormente entrenado, el árbol de decisión. No se puede destacar tampoco la performance del modelo con relación a los modelos anteriores

SVM

Comenzamos este modelo probando los tres kernels (lineal, polinómico y radial) para ver cual era mejor Kernel. Resultó ser el Kernel lineal y nos dio un f1_score de 0,75. Aun así decidimos probar por separado cada uno de los kernels. El lineal lo optimizamos pero no obtuvimos mejora en el score. El Kernel polinómico sin optimizar nos dio un score relativamente bajo (0,6). Al intentar optimizar sus hiperparametros, no pudimos terminar de correr el algoritmo debido al tiempo que tomaba. Finalmente el Kernel radial nos dio un f1_score muy bajo (0,60) y al optimizar hiperparametros apenas mejoró (0,67). Además para este caso no pudimos terminar de entrenar con CV para ver su capacidad de generalización.

Random Forest

Para el modelo random forest realizamos tres instancias. En la primera hicimos un modelo utilizando hiperparametros totalmente aleatorios. Para nuestra sorpresa, obtuvimos resultados bastante decentes (Un 80% de precisión en el set de testeo).

Después de esto decidimos optimizar los hiperparametros del random forest con validación cruzada usando grid search optimizando la f1 score. Estas optimizaciones implicaron una mejora del $\sim 3\%$ en el set de datos de testeos. Finalmente, decidimos hacer una optimización por validación cruzada con grid search buscando mejorar la mayor cantidad de métricas simultáneamente (precisamente: accuracy , f1 score, área bajo la curva roc, recall y precisión). A pesar de toda esta optimización, no hubo ninguna mejora considerable a la hora de predecir el set de testeo (solo hubo una mejora del ~ 0.01).

XGBoost

El modelo XGBoost representa el primer modelo que en su forma base genera la mejor predicción de todos los modelos entrenados en el análisis. Se realiza una búsqueda de los mejores hiperparametros, la cual, no genera una mejora consistente en las capacidades predictivas del modelo. Por lo tanto, se pueden llegar a las conclusiones que los modelos de ensambles usando árboles y los modelos de árboles son los más precisiones la hora de realizar predicciones

Voting

Tanto para el ensamble voting como stacking, decidimos usar los mejores modelos que obtuvimos en las etapas anteriores. Esto lo hicimos para poder sacar el mayor jugo de los modelos que habíamos obtenido antes.

Sin embargo, para nuestra sorpresa, obtuvimos resultados peores que con los modelos por separado. Para tratar mejorar este resultado, eliminamos el modelo svm (el cual nos había dado malos resultados) y cambiamos el modelo de votación a una votación ponderada. Esto nos dio resultados mejores, pero no superior al XGBoost

Stacking