Rapidminer – Ejercicios Varios

Modelling

Se crean modelos con diferentes algoritmos: árbol de decisión, naive bayes y rule induction. Se comparan los resultados y se analiza la forma visual en la que estos se muestran.

La forma en la que se muestran los modelos creados por ejemplo el árbol simplifica mucho el entendimiento del modelo generado.

Scoring

Se entrena un modelo utilizando naive bayes con el set de datos de entrenamiento y este se aplica a un set que no tiene la variable a predecir. El modelo aplica las predicciones.

Se puede ver que el modelo genera un grado de confianza para yes y otro para no. La decisión final para la predicción se define tomando el mayor de los 2.

Test Splits and validation

Se realiza un Split de los datos de entrenamiento de titanic con un ratio 70/30 train/test y se entrena un modelo utilizando el algoritmo naive bayes. Se obtiene una accuracy de 80% y una performance relativamente balanceada en cuanto a recall (72%) y precisión (75%), no hay ninguno de estos que haya sido optimizado por el modelo.

Cross validation

Se utiliza el bloque de cross validation que es un proceso en sí. Dentro de este, se entrenan los k modelos con árboles de decisión.

Se puede ver una accuracy de 80% +- 4% (desviación estándar) dependiendo del set elegido para test.

Se pueden utilizar los carets verdes situados dentro de parámetros de bloques para obtener información sobre qué opciones está utilizando la comunidad de rapidminer.

Comparación de algoritmos para el dataset de titanic

Árbol de decisión: 80.35% +/- 4.69%

Árbol con boosting de gradiente (no encontré XGB en rapiodminer): accuracy: 79.92% +/- 3.88%

Random forest: accuracy: 80.78% +/- 4.53%

Naive bayes: 78.17% +/- 4.83%

Red neuronal : tengo que transformar atributos binomiales y categoricos en one hot encoding (con bloque nominal to numerical -> el one hot encoding no andaba). 80.46% +/- 3.87%

Deep Learning: 78.61% +/- 5.51%

Resultados: Los mejores resultados son los del random forest. Inteersante la desviación estándar menor del argoritmo de boosting de gradiente. Se nota que la red neuronal profunda necesita más datos para obtener mejores resultados que los otros algoritmos de ML.

Visual Model Comparison

Se analiza la curva roc de 3 modelos diferentes entrenados con el mismo dataset pero diferentes algoritmos (naive bayes, árbol de decisión y rule induction). En este caso, el modelo de árbol fue el que mejor performo y el de naive bayes obtuvo la peor curva.