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

Arbol de desicion

Comenzamos este segundo checkpoint adaptando el dataset de testeo al Dataset de train con el que trabajamos en la primera parte. Esto consistió principalmente en adaptar el dataset de testeo al dataset de entrenamiento para sean compatibles con el modelo que creamos. En el camino tuvimos algunos registros con datos que no habíamos contemplado anteriormente (e incluso datos faltantes); estos los tuvimos que adaptar a nuestro dataframe de entrenamiento para usar el modelo de predicción. Luego aplicamos en ambos Datasets la técnica de One Hot Encoding para generar columnas numéricas correspondientes a las variables categóricas. Esto es necesario ya que los árboles de decisión que usamos en el tp sólo pueden trabajar con valores numéricos.

Una vez realizado el emparejamiento entre datasets de test y train, comenzamos a trabajar con árboles. Primero separamos el dataset de train hotels_train.csv en dos datasets: Uno lo utilizamos para entrenar nuestro modelo y el otro para testear el modelo obtenido. Elegimos una proporción de 80/20. Inicialmente, probamos con un árbol con una profundidad máxima de 20 y creamos un árbol utilizando el criterio Gini. Este modelo fue generado directamente tomando en cuenta todos los valores y sin generar ningún tipo de poda, para observar cómo se comporta el modelo sin hiper parámetros. Al graficarlo obtenemos, por supuesto, un árbol muy grande y extremadamente complejo. Al hacer nuestra primera predicción exportando el csv obtuvimos un score de 0,79 en la competencia de Kaggle (nada mal para un árbol sin ningún tipo de optimización).

Para mejorar nuestra predicción y encontrar un árbol más performante, procedimos a crear un nuevo árbol con el que trabajamos para mejorar sus hiperparametros. En éste, usamos 10 folds y probamos 15 combinaciones posibles entre los parámetros para buscar la mejor entre todas ellas. Para ellos se buscó la optimización del F1_score, el cual fue seleccionado debido a que es un buen equilibrio entre la precisión y el recall. Con este árbol obtuvimos un valor levemente mejor de F1_score. Sin embargo el mayor beneficio es que simplifica de manera significativa las reglas utilizadas para predecir y disminuye considerablemente el tamaño del árbol.

Por último, para evaluar el árbol obtenido utilizamos la técnica de Cross Validation con 10 folds. Así comprobamos que el modelo no cae en los fenómenos del overfitting ni underfitting.

Para concluir el análisis es importante destacar que en los múltiples entrenamientos obtuvimos resultados en un rango de 0.79 a 0.81 (con muy poca variación entre si). Esto nos indicaría que si bien el árbol de decisión es buen método de predicción; la mejora de sus hiper parámetros no implicó un aumento significativo del F1 score (independientemente de su tamaño).