Memoria Descriptiva

Predicción de grado de daño en estructuras en Machine Learning

En el presente proyecto de machine learning nos encontramos frente a un ejercicio de clasificación de variables, en donde se intentará determinar dadas una serie de características el grado de daño de la estructura en caso de un evento sísmico.

El dataset del presente proyecto contiene datos relativos al terremoto de Nepal del año 2015. La recolección de los mismo es hecha por el gobierno de Nepal, con el propósito de determinar las personas que necesitarían ayudas gubernamentales para la reconstrucción de sus hogares.

Por tanto, el dataset consta solo de edificios cuyo uso principal es la vivienda y algunos con usos secundarios. La encuesta principal consta de una serie de variables que fueron posteriormente purgadas para la competencia, dejando solo aquellas relacionadas con los edificios y el daño sufrido por los mismos, y resultando en el set de datos tratado en el proyecto.

El objetivo consiste en determinar 3 categorías que denominan el grado daño del edificio, desde daños superficiales que requieren reparaciones cosméticas (categoría 1), hasta daños importantes a la estructura que requerirían la reparación casi integral del inmueble (1 y 2).

Por otro lado, los atributos están constituidos por 38 columnas que describen diversos aspectos de cada edificio evaluado, los cuales podrían dividirse principalmente en 3 categorías: relacionados con el uso del edificio, características descriptivas del edificio y finalmente características con respecto a su ubicación geográfica.

La primera aproximación al modelo fue intentar determinar las variables más importantes en las predicciones, con el objetivo de sólo quedarnos con aquellas que interviniesen verdaderamente en la predicción y mejorar al mismo tiempo la eficiencia computacional.

Para ello, utilizamos algoritmos relativamente sencillos, el primer algoritmo implementado fue el ExtraTreeClassifiers, que determina la importancia de variables al igual que RandomForest o DecisionTrees, pero al ser aleatorio, es más rápido que estos últimos y reduce el tiempo de cálculo, importante para un set de datos grande en relación con la capacidad de cómputo disponible.

De la importancia de variables extraída se puede observar que todas aquellas relacionadas con la posición geográfica poseen mayor importancia. Esto podría explicar suponiendo que todas aquellas áreas geográficas cercanas al epicentro del terremoto serían las más afectadas y que el daño es menor conforme nos vamos alejando del epicentro. Asimismo la antigüedad, la edad y la altura del edificio resultan también importantes. En efecto, la altura de un edificio aumenta su centro de gravedad, lo cual en combinación con los materiales utilizados para la construcción, podrían aumentar la propensión a colapsar en caso de un evento sísmico, sobretodo aunado con estructuras rígidas y uniformes con pocos puntos flexibles.

Se observa también que todas las categorías que especifican los usos secundarios se ubican en las últimas posiciones tanto como para el RandomForest como para el ExtraTreeClassifier. Sabiendo que todos los edificios parte de la muestra tienen como uso principal vivienda, a menos que el uso secundario suponga un peso significativo en la estructura, el mismo no tiene influiría en el grado de daños observados.

Por otro lado, para poder alimentar a la mayoría de algoritmos las categorías que constan de clases codeadas como strings en numero íntegros, se intenta convertir dichos atributos con One Hot Encoding y Label Encoding, para observar el comportamiento de los algoritmos de predicción para cada uno. Esta conversión supuso un empeoramiento importante de los tiempos de cálculo de los algoritmos y ralentizó de manera importante el proceso de búsqueda y optimización.

En este sentido, en un intento de mejorar los tiempos de cálculo de los algoritmos, intentamos en primera instancia calcular una predicción simple con todas los atributos presentes, en este caso un RandomForest y un ExtraTreesClassifier, así como KNeighbors y XGBoost; para luego quedarnos con los algoritmos que alojaban mejores métricas, e intentamos mejorarlos tratando sus diferentes variables (profundidad, pureza de la hora, iteraciones, etc).

De este proceso obtenemos que RandomForest tiene mejor desempeño, y realizamos una primera submisssion en donde el score baja un poco para queda en 0.72.

Posteriormente, se intenta con CatBoostClassifier, el cual en primeras instancias sin modificar sus parámetros por defecto, arrojaba scores parecidos al Random Forest con el mejor score, a saber 0.71-0.73. Sin embargo, una vez los parámetros fueron ligeramente modificados, en especial la regularización y el numero de iteraciones, las métricas mejoran y se mantienen en valores constantes al realizar validación cruzada.

Por otro lado, una vez teniendo un score aceptable, se intenta reducir la dimensionalidad del modelo a través de un Embedding de las variables geográficas realizado con una red neuronal. De esta forma, la variable Geo Level 1 fue reducida de 30 a 16 categorías. Igualmente, eliminamos las columnas que aparecían en últimos lugares de las diversas importancia de variables y que además estaban relacionadas con los usos secundarios del edificio.

Sin embargo, al hacer la predicción en el nuevo dataset, el score se redujo significativamente con respecto al catboost previo, lo cual podría indicar que el Embedding fue realizado incorrectamente o con pocas dimensiones, o que que las variables que habían sido eliminadas, en conjunción con otras variables, eran importantes para realizar las predicciones. Finalmente nos quedamos con el modelos de Catboost de la 2da submission, a pesar que el modelo es altamente mejorable trabajando ciertos obstáculos como la distribución del target y la cantidad de variables en el set de datos.