

Machine Learning en modelos BIM

Núñez-Calzado, Pedro Enrique; Alarcón-López, Ivón José y Martínez-Gómez, David Carlos

Núñez-Calzado, Pedro Enrique – Arquitecto – BIM Expert – C# Developer – pnunez@ibim.es; Alarcón-López, Ivón José – Arquitecto Técnico – Consultor BIM – C# Developer – ivan@ibim.es; Martínez-Gómez, David Carlos – Arquitecto – Consultor BIM – david@ibim.es

Abstract

Machine learning (ML) is about creating software capable of extrapolating behaviour from information provided by examples, and enabling a computer to gradually learn.

The creation of a large or very detailed building information modelling (BIM) involves many processes in which users must enter information by hand.

This paper examines how the time needed for this task can be minimised by applying machine learning techniques to the design of a project with BIM. This enables the computer to assign values to parameters directly, according to initial guidelines and the learning process inherent in this technology.

Specifically, said techniques were used to automatically assign the names of rooms in the BIM of a housing development during the early design phases with considerable success.

Techniques of this type enable considerable time savings in comparison with manual or semiautomatic assignment procedures.

Keywords: *Machine Learning, Computational BIM, Efficiency, Revit API, C#.*

Resumen

Con el aprendizaje automático (del inglés, "Machine Learning") se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos, de forma que el ordenador aprende progresivamente.

En la construcción de un modelo BIM de cierto tamaño o de un nivel de definición elevado, hay muchos procesos en los que es necesaria la introducción de información de forma manual por parte del usuario.

Para minimizar el impacto de tiempo que supone este tipo de acciones, el artículo desarrolla el procedimiento de la aplicación de técnicas de Machine Learning a la fase de diseño de un proyecto con un modelo BIM, de forma que el ordenador asigna valores de parámetros directamente, según unas directrices iniciales y el propio proceso de aprendizaje inherente a esta tecnología.

En concreto, se aplica a la asignación automática del nombre de las habitaciones de un modelo BIM de un proyecto de un edificio residencial, en las fases iniciales de diseño, con un alto índice de acierto.

Este tipo de herramientas supone un ahorro de tiempo considerable respecto a un procedimiento de asignación manual o semiautomatizado.

Palabras clave: *Machine Learning, Automatización BIM, Aprendizaje automático, API Revit, C#.*

Introducción

El “Aprendizaje Automático” (del inglés, “Machine Learning”) es un subcampo de las ciencias de la computación y una rama de la inteligencia artificial, que dota a los ordenadores de la capacidad de “aprender”. Se crean algoritmos que son capaces de desarrollar respuestas lógicas a partir de datos, en contraposición al enfoque basado en programación explícita.

El “Modelado de información de construcción” (BIM, “Building Information Modeling”) es el proceso de generación y gestión de datos de un edificio durante su ciclo de vida. El modelo BIM abarca la geometría del edificio, las relaciones espaciales, la información geográfica, así como las cantidades y las propiedades de sus componentes.

El propósito de esta comunicación es tender un puente entre ambos campos (Machine Learning y BIM), mostrando un ejemplo en el que se aplican algoritmos de aprendizaje automático a los datos de un modelo BIM, generando como resultado final la predicción automática de nuevos datos.

El problema elegido para la aplicación del aprendizaje automático es la asignación del uso (salón comedor, cocina, dormitorio, etc.) a las dependencias de una vivienda. Usualmente, dicha asignación es totalmente manual por parte del modelador. A lo largo del artículo desarrollaremos la metodología para abordar el problema y evaluaremos la bondad de la solución propuesta.

Aunque la bibliografía y referencias acerca de la tecnología BIM es extensa, al igual que en el caso del Machine Learning, no hemos encontrado muchas referencias que versen sobre la aplicación de esta última al BIM, especialmente cuando se trata de realizar predicciones en modelos BIM a partir de otros modelos.

Este artículo es una aportación al desarrollo de la tecnología de los procesos de diseño en el ámbito del sector de la construcción, en el que pensamos que el Aprendizaje Automático tendrá un papel cada vez más relevante.

Los modelos de aprendizaje automático y gráficos de los datos que se han creado para esta publicación se han realizado mediante la herramienta web bigml[®], propiedad de BigML, Inc., una empresa estadounidense que ofrece servicios web de Machine Learning (<https://bigml.com/>).

1. Análisis previo

La asignación automática de los usos de las dependencias de una vivienda se encuadra dentro del denominado “aprendizaje supervisado”, en el que la base de conocimiento del sistema está formada por ejemplos etiquetados anteriores. Es decir, los datos de partida de los que aprenderá el algoritmo serán dependencias de la vivienda previamente etiquetadas con su uso correspondiente de forma manual.

Las técnicas que consideramos que mejor se adaptan a las premisas de este problema son el “Árbol de decisión”, la “Regresión Logística” y la “Red Neuronal”, que constituyen tres algoritmos diferentes de “Clasificación”. Sus características fundamentales son:

- **Árbol de decisión:** el algoritmo estructura los datos de partida en forma de árbol con condiciones en cada rama, de forma que cada nodo constituye una predicción basada en las condiciones impuestas por las ramas recorridas.
- **Regresión Logística:** es similar a la clásica regresión lineal, solo que en este caso la predicción resultado es un valor (probabilidad) entre 0 y 1. Para ello se hace uso de la función logística, cuya curva es en forma de S.
- **Red Neuronal:** se basa en una estructura similar a las neuronas cerebrales, en la que existen umbrales de activación de las neuronas. Se crea una estructura compleja con una entrada de datos múltiple, una salida de datos múltiple y una o varias capas ocultas de procesado.

2. Datos de partida

Los datos con los que “entrenar” nuestro algoritmo son una pieza fundamental del puzzle. En este caso, los datos van a provenir de dos modelos de vivienda colectiva realizados con Autodesk Revit®, de más de 200 viviendas cada uno, de los que se extraerán parte de las habitaciones. Los algoritmos se han de entrenar con datos diferentes a los datos con los que se evalúa su rendimiento, para que los resultados de la evaluación sean significativos.

Entre las dos opciones de que disponemos (crear un conjunto de datos completo y dividirlo aleatoriamente entre datos de entrenamiento y datos de evaluación, o usar un modelo de viviendas como datos de entrenamiento y el otro como datos de evaluación) optamos por la opción más conservadora, que es entrenar el algoritmo con un proyecto y evaluarlo con otro diferente. De esta forma, podemos asegurar que los resultados son extrapolables a otros proyectos.

Tal y como se estructura la información de los modelos en Autodesk Revit®, existen entidades “muro”, “habitación” y “puerta”. Las entidades “habitación” poseen información acerca de la geometría de las dependencias del modelo generada a partir de los “muros”, y las entidades “puerta” permiten conocer las conexiones entre habitaciones.

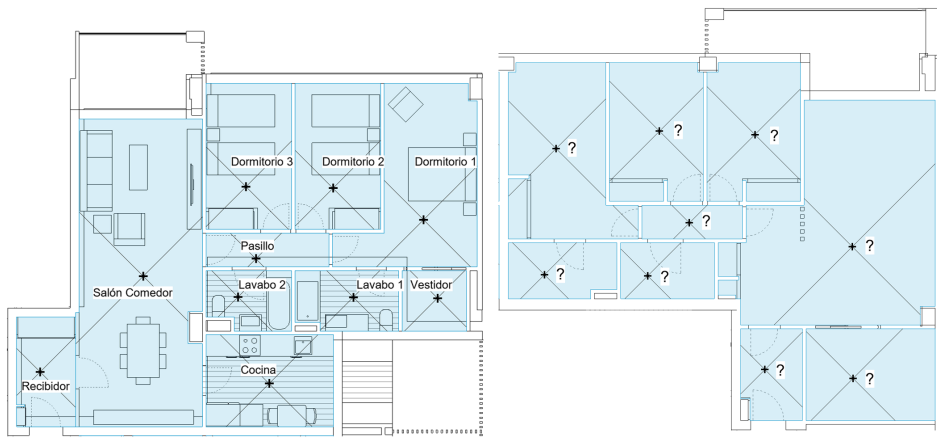


Fig. 1. A la izquierda, una de las viviendas del proyecto de entrenamiento. A la derecha, una vivienda a predecir por el algoritmo ya entrenado. Fuente: elaboración propia (2018).

La decisión de qué elementos han de modelarse para poder realizar predicciones supone un compromiso entre calidad de predicción y coste de adquisición de datos. En nuestro caso, y tras diversas pruebas previas, concluimos que para realizar una predicción razonablemente buena debíamos modelar como mínimo la tabiquería (muros), las puertas y las habitaciones.

Partiremos por tanto de un modelo con muros y puertas modelados, en el que se han generado habitaciones para las dependencias de las viviendas. Las habitaciones tendrán un parámetro que contiene su uso y cuya asignación se ha realizado manualmente. Para esta investigación hemos tomado la decisión de simplificar los usos a los más comunes: recibidor, salón comedor, cocina, pasillo, dormitorio y lavabo. Por tanto, se han eliminado los datos de usos más accesorios (terrazas, despensas y vestidores) para poder compatibilizar al máximo los datos con los que contamos.

Una transformación de los datos que conviene destacar, es que es necesario eliminar la numeración tradicional (dormitorio 1, dormitorio 2, etc.) de los datos extraídos, puesto que a efectos del algoritmo, la etiqueta ha de ser “dormitorio”, que es el “tipo” de estancia. Esto también significa que el algoritmo solo podrá predecir “dormitorio” en este caso, con lo que tendremos que realizar un postproceso automático de reenumeración de las estancias con varios ejemplares por vivienda una vez concluida la predicción.

Esta transformación puede realizarse una vez extraídos los datos, o incluso podría realizarse en el propio Revit®, si el uso se asigna a través de una tabla de claves, asignando simultáneamente otro parámetro destinado al “tipo de estancia”.

Otro aspecto importante en la elección de los datos de partida es su extensión, es decir, la cantidad de información que se extrae de los modelos. Con el fin de comparar la sensibilidad en cuanto a la extensión de la información que se posee, hemos creado dos conjuntos de datos diferentes para cada modelo. En el primero solo tenemos disponible la información que se obtiene directamente (o mediante fórmulas) en las tablas de planificación de Revit®. En el segundo añadimos información que solo puede extraerse o calcularse mediante programación en C# con la API (Application Programming Interface) de Revit® o medios similares como Dynamo®.

Ejemplo de información extraída de una sola vivienda. Fuente: elaboración propia (2018).

Etiqueta	Datos tabla planificación				Datos extraídos mediante la API de Revit®									
Código	Área (m ²)	Perímetro (m)	Volumen (m ³)	Cuadratura	Máximo caja (m)	Mínimo caja (m)	Min / Max caja	Número lados	Área / Área caja	Jerarquía	Orden por Área	Dependientes	Habitaciones Vivienda	Área / Área Vivienda
SalonComedor	20,25	20,00	50,63	0,9002	6,40	3,60	0,5622	7	0,8795	1	0	0	9	0,2578
Dormitorio	13,80	19,94	34,50	0,7451	5,20	3,67	0,7062	18	0,7227	2	1	1	9	0,1757
Dormitorio	9,79	14,10	24,46	0,8874	4,10	2,40	0,5854	10	0,9944	2	2	0	9	0,1246
Dormitorio	9,70	14,10	24,26	0,8837	4,10	2,40	0,5854	13	0,9861	2	3	0	9	0,1235
Cocina	8,94	12,60	22,34	0,9490	4,10	2,20	0,5366	9	0,9907	1	4	0	9	0,1138
Recibidor	5,62	10,70	14,04	0,8858	2,70	2,10	0,7778	11	0,9903	0	5	8	9	0,0715
Lavabo	3,76	7,90	9,41	0,9819	2,35	1,60	0,6813	6	1,0000	3	6	0	9	0,0479
Lavabo	3,52	7,60	8,81	0,9875	2,20	1,60	0,7277	5	1,0000	2	7	0	9	0,0448
Pasillo	3,17	8,35	7,94	0,8532	3,18	1,00	0,3144	8	1,0000	1	8	5	9	0,0404

La definición de los parámetros extraídos y calculados para cada habitación es la siguiente:

- Código: es el valor del parámetro que corresponde al uso de la habitación.
- Área (m²): es la superficie de la habitación en metros cuadrados.
- Perímetro (m): es el perímetro de la habitación en metros.
- Volumen (m³): es el volumen de la habitación en metros cúbicos.
- Cuadratura: es un factor que mide lo similar que es la habitación a un cuadrado. Su valor oscila entre 0 y 1. Se calcula como la raíz cuadrada del área, multiplicada por 4 y dividida por el perímetro.
- Máximo caja (m): es la dimensión máxima del rectángulo más pequeño que puede contener la geometría de la habitación.
- Mínimo caja (m): es la dimensión mínima del rectángulo más pequeño que puede contener la geometría de la habitación.
- Min / Max caja: es el cociente entre el “mínimo caja” y el “máximo caja”. Oscila entre 0 y 1.
- Numero lados: número de entidades diferentes que dan lugar al contorno de la habitación.
- Área / Área caja: es el cociente entre el área de la habitación y el área del rectángulo más pequeño que puede contener la geometría de la habitación.
- Jerarquía: es el número de habitaciones que hay que atravesar desde la entrada de la vivienda para llegar a la habitación.

- Orden por Área: es el número que ocupa dentro de la lista de habitaciones de la vivienda ordenadas por área de mayor a menor, comenzando en 0 para la habitación más grande.
- Dependientes: es el número de habitaciones al que se puede llegar desde la habitación y por las que no se ha pasado previamente para llegar a ella.
- Habitaciones Vivienda: es el número total de las dependencias de la vivienda a la que pertenece la habitación.
- Área / Área Vivienda: es el cociente entre el área de la habitación y el área de la vivienda a la que pertenece. Oscila entre 0 y 1.

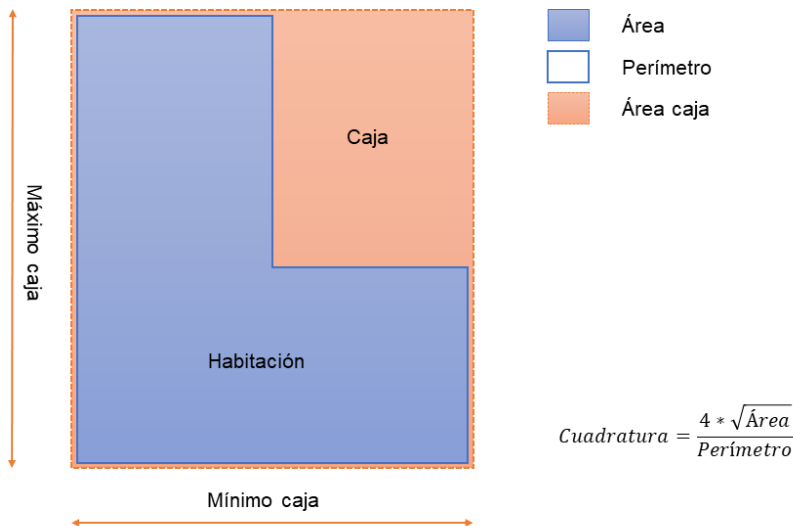


Fig. 2. Dibujo explicativo de los parámetros de una habitación. Fuente: elaboración propia (2018).

Dentro de los parámetros que se pueden obtener mediante tablas de planificación, la “Cuadratura” se calcula mediante una fórmula, estando los demás disponibles directamente en las habitaciones.

Con respecto a los datos que se han extraído mediante la API de Revit®, se ha recurrido a dos estrategias para obtenerlos. Primeramente, se obtiene la geometría de la caja envolvente de la habitación (rectángulo más pequeño que puede contener la geometría de la habitación), con lo que podemos calcular todos los parámetros relativos a ésta.

Posteriormente, tomando como base la información de las puertas (la puerta de entrada a la vivienda se distingue con un parámetro especial), somos capaces de identificar todas las unidades de vivienda y de reconstruir el árbol completo de habitaciones y conexiones de cada una, lo que nos permite generar el resto de los parámetros extraídos.

3. Importación de datos inicial

Tras la extracción de datos desde Autodesk Revit®, obtenemos dos archivos en formato CSV, uno de cada proyecto. Los archivos CSV (del inglés, comma-separated values) son un tipo de documento sencillo de formato texto para representar los datos en forma de tabla, en que las columnas se separan por comas o puntos y coma, y las filas por saltos de línea.

Estos datos se convierten en cuatro conjuntos de datos (dos completos y dos eliminando las columnas creadas mediante la API de Revit®). Todos constan de varios campos numéricos y un campo categórico, que se corresponde con el uso de la habitación.

Es muy importante realizar una revisión de los datos para detectar omisiones, incoherencias y errores. En nuestro contexto, podemos decir que es mejor eliminar una fila de datos (habitación), que mantener una fila con errores. La premisa fundamental es la conocida como GIGO (garbage in, garbage out), es decir, que si introducimos unos datos de entrada erróneos, no podemos esperar una predicción acertada.

Resumen de los datos importados. Fuente: elaboración propia (2018).

Conjunto de datos	Proyecto	Campos		Filas
		Núméricos	Catagóricos	
1	Proyecto 1	14	1	590
2	Proyecto 2	14	1	912
3	Proyecto 1	4	1	590
4	Proyecto 2	4	1	912

En este momento, podemos realizar alguna visualización de los datos, la cual, aunque ha de ser parcial (no se puede generar un gráfico razonable a partir de más de 4 o 5 variables), puede ayudarnos a conocer con más profundidad la estructura de los datos de que disponemos.

En las figuras Fig. 3 y Fig. 4 se muestran dos gráficos que nos parecen representativos de la capacidad que tienen los parámetros elegidos de diferenciar el uso de las dependencias. En ambos casos el uso de la habitación se representa mediante una leyenda de colores, mientras que tanto en ordenadas como en abcisas se especifica alguno de los parámetros numéricos. En este tipo de gráfico, cuanto más agrupados veamos los círculos del mismo color, significa que los parámetros que hemos elegido tienen mayor poder predictivo para clasificar el uso de la habitación.

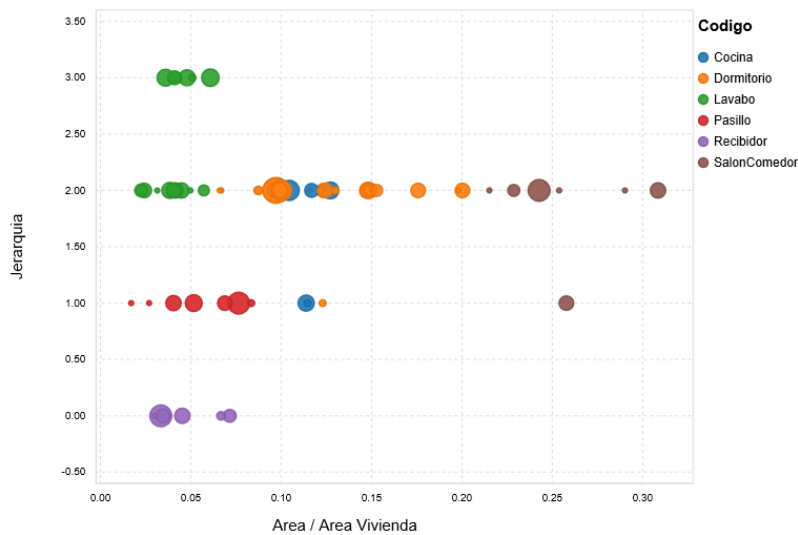


Fig. 3. Gráfico de las habitaciones del proyecto 2 según Área / Área Vivienda y Jerarquía. Fuente: elaboración propia (2018).

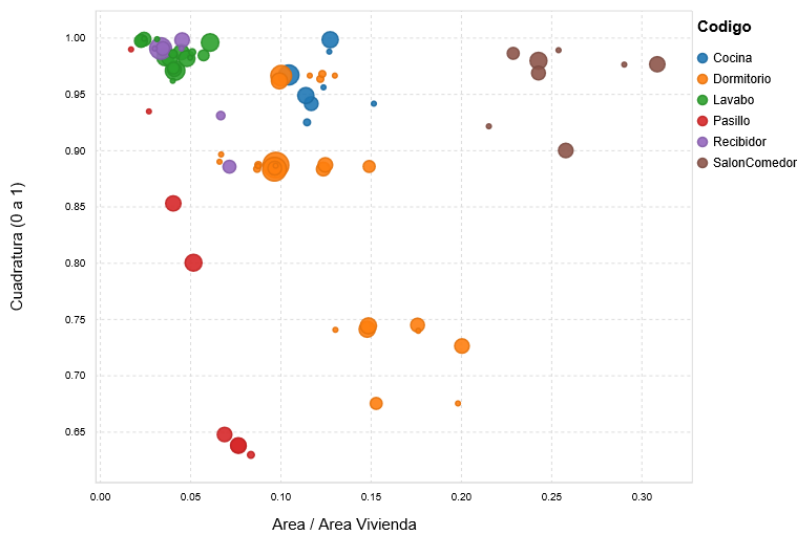


Fig. 4. Gráfico de las habitaciones del proyecto 2 según Área / Área Vivienda y Cuadratura. Fuente: elaboración propia (2018).

4. Entrenamiento de los algoritmos

Una vez extraídos los datos, podemos entrenar los diferentes algoritmos con ellos y posteriormente acceder a los resultados del proceso de entrenamiento. Básicamente, cada algoritmo nos muestra cuál es su “forma de ver” los datos que le hemos introducido. La visualización que se obtiene en el caso de la Regresión Logística y la Red Neuronal es bastante parcial y no da una idea global de cómo estructura los datos el algoritmo. Sin embargo, en el caso del Árbol de decisión, la visualización es bastante descriptiva.

- **Árbol de decisión:** en este caso, podemos consultar el árbol completo (o parte de él), y situándonos en cada nodo podremos apreciar la ruta completa a través de las ramas, con su traducción en condiciones sobre los parámetros, y su predicción final de la clasificación del nodo. La clasificación de cada nodo tiene asociado un nivel de confianza:

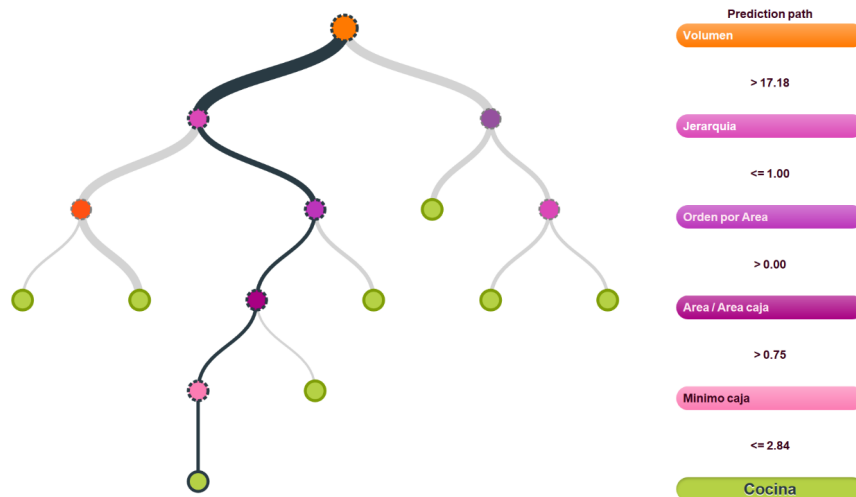


Fig. 5. Gráfico ejemplo de Árbol de Decisión. Fuente: elaboración propia (2018).

- **Regresión Logística:** como resultado del entrenamiento, podemos ver gráficas que nos muestran la probabilidad de cada uso según la variación de uno de los parámetros (en este caso el Área), manteniendo el resto de parámetros como constantes con valores elegidos por el usuario.

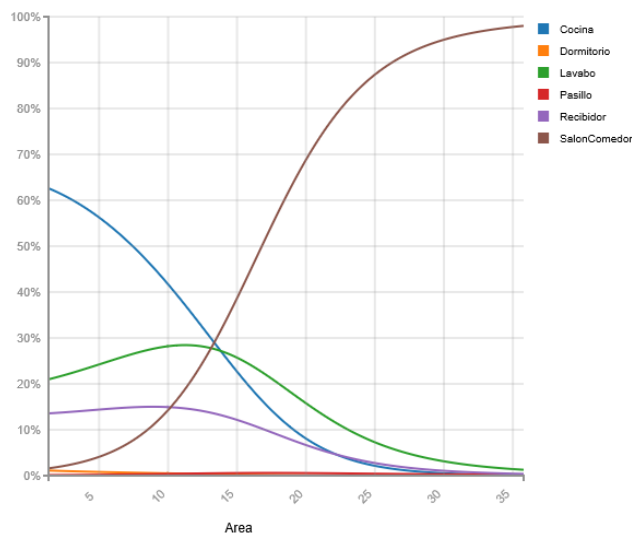


Fig. 6. Gráfico ejemplo de resultados parciales de Regresión Logística. Fuente: elaboración propia (2018).

- **Red Neuronal:** en este caso, el gráfico nos muestra el uso en función de dos de los parámetros (jerarquía y cuadratura), manteniendo el resto como constantes con valores elegidos por el usuario.

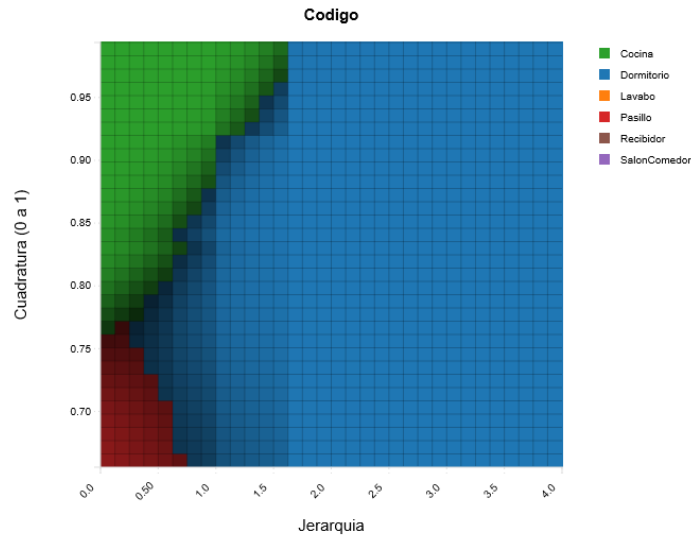


Fig. 7. Gráfico ejemplo de resultados parciales de Red Neuronal. Fuente: elaboración propia (2018).

5. Evaluación de los algoritmos

El siguiente paso dentro del proceso es la evaluación de la capacidad predictiva de los algoritmos ya entrenados. Para ello, hacemos una predicción con el algoritmo sobre el conjunto de datos de evaluación (como si no estuviese ya asignado el resultado) y posteriormente comparamos la predicción con los valores reales asignados.

En nuestro caso concreto, para evaluar el algoritmo entrenado con los datos del proyecto 1, usaremos el algoritmo para predecir los usos de las habitaciones del proyecto 2 y compararemos la predicción obtenida con la asignación real de las habitaciones del proyecto 2.

Como resultado de la evaluación de cada algoritmo, obtendremos una “matriz de confusión” en la que se establece la comparativa entre la predicción y la realidad. A la derecha de la tabla se aprecian los coeficientes F, que van de 0 a 1 y miden la calidad de la predicción. Los aciertos están marcados en verde y los errores en rojo.

Matriz de Confusión ejemplo (MED se refiere a Media). Fuente: elaboración propia (2018).

REAL \ PREDICCIÓN	Cocina	Dormitorio	Lavabo	Pasillo	Recibidor	SalonComedor	TOTAL REAL	EXHAUSTIVIDAD	F
Cocina	55	13	0	0	0	0	68	80,88%	0,8462
Dormitorio	7	188	14	0	0	2	211	89,10%	0,9126
Lavabo	0	0	136	0	0	0	136	100,00%	0,9252
Pasillo	0	0	8	60	0	0	68	88,24%	0,9302
Recibidor	0	0	0	1	38	0	39	97,44%	0,9870
SalonComedor	0	0	0	0	0	68	68	100,00%	0,9855
TOTAL PREDICCIÓN	62	201	158	61	38	70	590	EXH.MED 92,61%	F.MED 0,9311
PRECISION	88,71%	93,53%	86,08%	98,36%	100,00%	97,14%	PRE.MED 93,97%	EXACTITUD 92,37%	

La definición de los parámetros usados en la matriz de confusion es la siguiente:

- VP: verdadero positivo.
- VN: verdadero negativo.
- FP: falso positivo.
- FN: falso negativo.
- $PRECISION = VP / (VP + FP)$
- $EXHAUSTIVIDAD = VP / (VP + FN)$
- $F = 2 * (PRECISION * EXHAUSTIVIDAD) / (PRECISION + EXHAUSTIVIDAD)$
- $EXACTITUD = (VP + VN) / (VP + FP + VN + FN)$

Finalmente, se han evaluado los dos proyectos, con datos parciales (provenientes de tablas de planificación) y completos, para los tres algoritmos seleccionados.

Resultados de la evaluación de múltiples algoritmos con los diferentes conjuntos de datos. Fuente: elaboración propia (2018).

DATOS	ALGORITMO	ENTRENAMIENTO	EVALUACIÓN	F	EXACTITUD
Completos	Árbol de Decisión	Proyecto 1	Proyecto 2	0,7248	82,0%
Completos	Red Neuronal	Proyecto 1	Proyecto 2	0,9113	93,6%
Completos	Regresión Logística	Proyecto 1	Proyecto 2	0,8122	83,2%
Completos	Árbol de Decisión	Proyecto 2	Proyecto 1	0,7728	81,4%
Completos	Red Neuronal	Proyecto 2	Proyecto 1	0,8497	84,9%
Completos	Regresión Logística	Proyecto 2	Proyecto 1	0,9311	92,4%
Parciales	Árbol de Decisión	Proyecto 1	Proyecto 2	0,7930	80,9%
Parciales	Red Neuronal	Proyecto 1	Proyecto 2	0,6808	72,4%
Parciales	Regresión Logística	Proyecto 1	Proyecto 2	0,6709	81,4%
Parciales	Árbol de Decisión	Proyecto 2	Proyecto 1	0,6630	74,1%
Parciales	Red Neuronal	Proyecto 2	Proyecto 1	0,7412	77,5%
Parciales	Regresión Logística	Proyecto 2	Proyecto 1	0,6027	80,2%

Resultados medios de la evaluación los diferentes algoritmos y conjuntos de datos. Fuente: elaboración propia (2018).

DATOS	ALGORITMO	Promedio de F	Promedio de EXACTITUD
Completos	Árbol de Decisión	0,7488	81,7%
	Red Neuronal	0,8805	89,3%
	Regresión Logística	0,8717	87,8%
Total Completos		0,8337	86,3%
Parciales	Árbol de Decisión	0,7280	77,5%
	Red Neuronal	0,7110	75,0%
	Regresión Logística	0,6368	80,8%
Total Parciales		0,6919	77,8%

En las tablas anteriores resaltamos la importancia del valor F (también denominado F-Score o F-Measure), que es la media armónica entre la PRECISION y la EXHAUSTIVIDAD, y nos ofrece una medida bastante objetiva del buen desempeño del algoritmo. Su valor oscila entre 1, que significa una predicción perfecta y 0, que es una predicción totalmente errónea.

Como podemos apreciar, con los datos completos se realizan mejores predicciones que con los datos provenientes únicamente de las tablas de planificación, por lo que merece la pena haber programado una extracción más extensa de datos con la API de Revit®. Con respecto a los algoritmos, tanto la Red Neuronal

como la Regresión Logística mejoran significativamente al introducir más columnas de datos, no así el Árbol de decisión, que apenas mejora su rendimiento.

Se aprecia una cierta asimetría entre los proyectos, dado que en el proyecto 1 funciona aparentemente mejor la Red Neuronal y en el proyecto 2 la Regresión Logística. Para comprobar con mayor fiabilidad qué algoritmo es el mejor, deberíamos disponer de más proyectos con los que evaluarlos.

6. Uso de los algoritmos

Una vez evaluado el desempeño de los algoritmos, éstos pueden usarse para asignar (predecir) el parámetro de uso de las habitaciones en un modelo en el que todavía no está asignado, sabiendo de antemano el nivel aproximado de exactitud total que podemos esperar.

El proceso resumido sería el siguiente:

- Modelar muros, puertas y habitaciones (sin asignación del uso).
- Realizar la extracción de datos de las habitaciones con la misma técnica que hemos empleado para los datos de entrenamiento del algoritmo, excepto el Código (uso), que es el campo a predecir. El proceso de extracción incluye el cálculo de las unidades de vivienda y la asignación de las habitaciones a cada unidad como efecto colateral beneficioso.
- Con el algoritmo de elección ya entrenado, realizar la predicción del campo Código.
- Importar los resultados de la predicción en Revit® mediante un plugin, la API o Dynamo®, para asignarlos al parámetro correspondiente de las habitaciones.
- Revisar y corregir los errores manualmente si es necesaria una exactitud mayor que la proporcionada por el algoritmo.

El hecho de que tengamos una estimación de exactitud que proviene de la evaluación del algoritmo no necesariamente quiere decir que esa es la exactitud que vamos a conseguir en nuevos proyectos, aunque lo normal es que no se produzcan variaciones drásticas.

A la vista de los resultados que hemos obtenido, y para ofrecer una intuición más concreta de cómo han funcionado los algoritmos, podríamos decir que si realizamos la predicción mediante una Regresión Logística o una Red Neuronal con los datos completos, podríamos esperar un porcentaje total de acierto de entre el 80% y el 90%. A la hora de revisar manualmente los resultados, los errores más esperados son la confusión de cocinas con dormitorios (en ambos sentidos) y la catalogación de pasillos o dormitorios como lavabos.

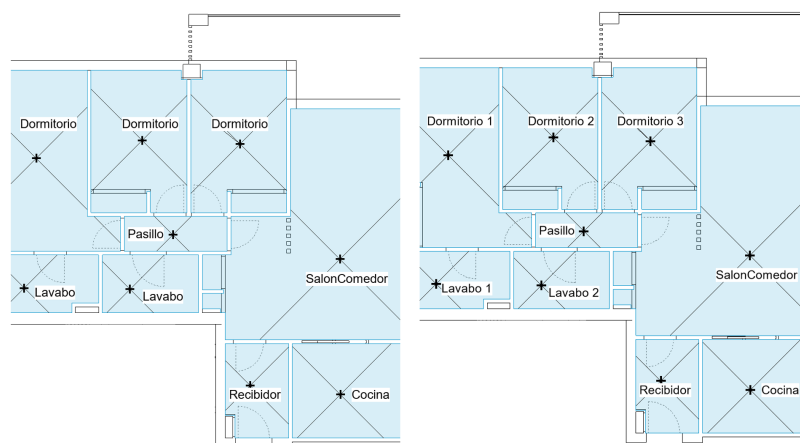


Fig. 8. Modelo de vivienda en Revit. A la izquierda, propuesta de nomenclatura de habitaciones por el algoritmo de regresión logística y a la derecha, valores finales asignados mediante programación adicional. Fuente: elaboración propia (2018).

En función de la tecnología usada para crear los modelos de predicción, entendemos que se podría llegar a integrar completamente la predicción del uso de habitaciones dentro de Revit® mediante la API.

7. Conclusiones

Las técnicas de Aprendizaje Automático (Machine Learning) poseen un importante potencial de aplicación en el desarrollo de modelos BIM. La aplicación de algoritmos de clasificación como los descritos permite automatizar tareas que de otra forma habrían de realizarse manualmente.

El empleo generalizado de estas herramientas, en nuestra opinión, requeriría adoptar un enfoque de los parámetros de los modelos más probabilístico y menos determinista, la adopción de medidas extraordinarias de corrección y postproceso, o ambos. Estas medidas podrían contemplar la introducción de una comprobación automática, que a partir de criterios establecidos con programación tradicional permita, al menos, detectar las viviendas en que el algoritmo ha cometido errores.

Además de la correcta elección de los algoritmos de predicción, la elección, cantidad y calidad de los datos con los que se van a entrenar estos algoritmos es la pieza clave para obtener resultados consistentes y con el suficiente potencial predictivo.

8. Referencias

- ALPAYDIN E. (2016). *Machine Learning: The New AI (The MIT Press Essential Knowledge series)*. USA: The MIT Press.
- BIGML. *Machine Learning made easy, beautiful and understandable*. <<https://bigml.com/>> [Consulta: 9 de marzo de 2017]
- FOREMAN J.W. (2013). *Data Smart: Using Data Science to Transform Information into Insight*. USA: Wiley.
- MURPHY K.P. (2012). *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. USA: The MIT Press.
- NG ANDREW. *Stanford University Machine Learning course*. <<https://es.coursera.org/learn/machine-learning>> [Consulta: 9 de marzo de 2017]
- PROVOST, F. y FAWCETT T. (2013). *Data Science for Business: What you need to know about data mining and data-analytic thinking*. USA: O'Reilly Media.
- REVIT. *About rooms*. <<http://help.autodesk.com/view/RVT/2018/ENU/?guid=GUID-DD74A51D-A0B0-4461-A4BA-0F9CCC191CDB>> [Consulta: 9 de marzo de 2017]
- REVIT. *Room instance properties*. <<http://help.autodesk.com/view/RVT/2018/ENU/?guid=GUID-21326970-0037-41C6-A996-980C24EE019F>> [Consulta: 9 de marzo de 2017]
- SHALEV-SHWARTZ S. y BEN-DAVID S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press
- SILVER N. (2014). *La señal y el ruido: Cómo navegar por la maraña de datos que nos inunda, localizar los que son relevantes y utilizarlos para elaborar predicciones infalibles*. España: Ediciones Península.