Predicción de ocupación de aulas con sensores ambientales (sin visión)

Dylam Joseph Jaime Guiza
Elmar Wilson Leguizamón Ballen
Yoiber Andrés Beitar Rentería
David Franchesco Rodríguez Celemín

dylamjosephj@utadeo.edu.co; elmarleguizamonb@utadeo.edu.co;
yoiberbeitarr@utadeo.edu.co; davidf.rodriguezc@utadeo.edu.co
Universidad Jorge Tadeo Lozano — Programa de Ingeniería de Sistemas
Curso: Inteligencia Artificial — Bogotá D.C., Colombia

05 de octubre de 2025

Resumen

Proponemos un proyecto de aprendizaje automático para **predecir la ocupación de aulas** exclusivamente a partir de **sensores ambientales** (temperatura, humedad, luz y CO2), sin utilizar imágenes ni audio. El enfoque reduce complejidad, costos y riesgos de privacidad, y es replicable en diferentes salones del campus. Utilizaremos el *Occupancy Detection Data Set* del repositorio UCI (con espejo en Kaggle), que contiene mediciones minuto a minuto en formato CSV y etiqueta binaria de ocupación. Entrenaremos y compararemos tres modelos: *Regresión Logística* (línea base interpretable), *Random Forest* (no lineal y robusto) y *SVM lineal*. Se evaluará con F1, ROC–AUC, Precisión, Recall y matriz de confusión bajo una **partición temporal** (train/val/test 70/15/15) para evitar fuga de información. Nuestra hipótesis es alcanzar **F1 > 0.90** y **AUC > 0.95** con Random Forest, entregando además código reproducible y repositorio Git con licencia abierta.

Problema local y motivación

En la Universidad Jorge Tadeo Lozano, la planificación de aulas depende de horarios teóricos y observaciones manuales, lo que impacta la *eficiencia* en el uso de espacios y el *consumo energético* (iluminación y ventilación). Una detección automatizada de ocupación permite liberar o reasignar ambientes en tiempo casi real, reducir costos e incorporar criterios de sostenibilidad. A diferencia de soluciones basadas en cámaras, el uso de sensores *no captura datos personales*, simplifica la operación y facilita su adopción por áreas administrativas y de TI del campus.

Dataset

- **Nombre y fuente:** *Occupancy Detection Data Set* (UCI, 2016) con espejo en Kaggle [1, 2, 3].

- **Tamaño y variables:** ∼20,560 instancias; variables: `Temperature`, `Humidity`, `Light`, `CO2`, `HumidityRatio`, y atributos derivados de fecha-hora.
- **Formato y licencia:** CSV con licencia abierta (CC0 en Kaggle) para uso educativo e investigación.
- **Validez:** Conjunto ampliamente usado en la literatura, con señales físicas robustas y representativas de oficinas/aulas en entornos reales.

Tarea de IA y algoritmo(s)

Tipo de datos: tabulares (sensores). **Tarea:** clasificación binaria *ocupado* vs. *vacío*.
Modelos propuestos:

- **Regresión Logística** (baseline): interpretable, rápida y con buen desempeño cuando la frontera es aproximadamente lineal [4, 5].
- **Random Forest** (principal): maneja no linealidades e interacciones sin ingeniería de características compleja; robusto a ruido y escalado [6].
- **SVM lineal** (comparativo): buen rendimiento en espacios de alta dimensión y con regularización explícita [7].

Metodología y evaluación

Preprocesamiento. Limpieza de nulos; *parsing* de fecha-hora; ingeniería ligera (hora del día, indicador fin de semana); estandarización para modelos lineales.

Partición temporal. Se aplica **train/val/test 70/15/15** manteniendo el orden cronológico para evitar *data leakage*. La validación guía la selección de hiperparámetros y umbrales.

Entrenamiento. Búsqueda acotada de hiperparámetros con validación cruzada temporal; balanceo de clases mediante `class_weight` o sobremuestreo si procede [8]. Se registran semillas y versiones para reproducibilidad.

Métricas. Reporte de Precisión, Recall, **F1**, **ROC–AUC** y matriz de confusión; curvas ROC y Precisión–Recall para el mejor modelo [9].

Líneas base y comparación. Compararemos Logistic vs. RF vs. SVM; una regla mayoritaria sirve como piso de desempeño mínimo.

Stack técnico. Python 3.10, `pandas/numpy`, `scikit-learn` y `matplotlib` [4].

Resultados esperados e hipótesis

- **Hipótesis 1:** *Random Forest* alcanzará **F1 $>$ 0.90** y **AUC $>$ 0.95** en el conjunto de prueba independiente.
- **Hipótesis 2:** Las variables `Light` y `CO2` figurarán entre las más relevantes en la importancia de características de RF.
- **Hipótesis 3:** La inclusión de *features* temporales (hora, fin de semana) mejorará F1 respecto a usar sólo señales físicas.

Consideraciones éticas y riesgos

El proyecto **no usa imágenes ni audio**, por lo que minimiza exposición a datos personales. Riesgos: sesgo por ubicación de sensores, deriva temporal en horarios y cambios de ventilación. Mitigaciones: calibración básica, evaluación por periodos y registro de supuestos. Se documentarán límites de generalización y buenas prácticas de despliegue responsable [10].

Alcance y cronograma

- **Semana 1:** EDA, preprocesamiento, definición de *features*.
- **Semana 2:** Entrenamiento y validación; selección de modelo.
- **Semana 3:** Evaluación final y análisis de errores; curvas ROC/PR; explicabilidad (importancias).
- **Semana 4:** Documento LaTeX, README y publicación del repositorio con `LICENSE`.

Roles del equipo

- **Dylam Joseph Jaime Guiza:** liderazgo técnico, modelado y validación.
- **Elmar Wilson Leguizamón Ballen:** preprocesamiento, métricas y documentación de resultados.
- **Yoiber Andrés Beitar Rentería:** LaTeX, README y estructura del repositorio Git.

Repositorio del proyecto: `https://github.com/organizacion/proyecto-ocupacion-sensores` (placeholder, sustituir por el repositorio real).

Referencias

[1] UCI Machine Learning Repository. *Occupancy Detection Data Set*. `https://archive.ics.uci.edu/dataset/357/occupancy+Detection`. Accedido: octubre 2025. 2016.

[2] Kaggle Datasets. *Room Occupancy Detection (IoT Sensor) - CC0 License*. `https://www.kaggle.com/datasets/kukuroo3/room-occupancy-detection-data-iot-sensor`. Accedido: octubre 2025. 2019.

[3] Luis M. Candanedo y Véronique Feldheim. «Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models». En: *Energy and Buildings* 112 (2016), págs. 28-39.

[4] Fabian Pedregosa et al. «Scikit-learn: Machine Learning in Python». En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.

[5] Ron Kohavi. «A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection». En: *IJCAI*. 1995, págs. 1137-1145.

[6] Leo Breiman. «Random Forests». En: *Machine Learning* 45.1 (2001), págs. 5-32.

[7] Corinna Cortes y Vladimir Vapnik. «Support-Vector Networks». En: *Machine Learning* 20.3 (1995), págs. 273-297.

[8] Haibo He y Edward A. Garcia. «Learning from Imbalanced Data». En: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), págs. 1263-1284.

[9] Takaya Saito y Marc Rehmsmeier. «The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets». En: *PLOS ONE* 10.3 (2015), e0118432.

[10] Chen Wang, Tao Huang y Kai Zhang. *Integrated Sensor Data Processing for Occupancy Detection*. Inf. téc. National Institute of Standards y Technology (NIST), 2021.