Universidad Francisco Marroquín
Christian Medina Armas
Machine Learning Models

Proyecto #1 -
Room Occupancy Estimation

Stephanie Grotewold
20210567
5 de marzo del 2024

Introducción

El propósito de este informe es estimar el número preciso de ocupantes en una habitación utilizando datos de múltiples sensores medioambientales no intrusivos. El conjunto de datos, recopilado en un experimento controlado, incluye información de sensores como temperatura, luz, sonido, CO2 y PIR (Infrarrojo Pasivo). El informe busca proporcionar información sobre los patrones de ocupación basados en datos de sensores.

Data

El conjunto de datos es multivariado y de series temporales, con 10,129 instancias y 18 características. El área temática es Ciencias de la Computación, y la tarea asociada es la Clasificación. Las características son valores reales, y los datos se recopilaron en una habitación de 6m x 4.6m utilizando un banco de pruebas con 7 nodos de sensores y un nodo central. Los sensores incluyen temperatura (C), luz(LUX), sonido(Volts), CO2 (PRM) y PIR. Se realizaron calibraciones manuales para los sensores de CO2, sonido y PIR.

Métodos

Análisis de Datos, Visualización y Análisis de normalidad

Al revisar el dataset, se pudo notar que incluía fechas, sin embargo, al ser solo datos recolectados de 4 días, no agregaba estacionalidad o información que se considera necesario. A pesar de eso, sí se tomó en cuenta la hora, ya que se encontró un patrón. Se notó que no faltaba ninguna observación y, por lo tanto, no se realizó ninguna imputación.

Generamos visualizaciones, incluyendo histogramas y boxplots, destacando concentración uniforme de temperaturas (25.5 °C - 26 °C) en sensores 1 y 4, variabilidad bimodal en la luz (sensores 1 y 4) y distribución bimodal en PIR. Los gráficos QQ confirmaron casi normalidad en temperaturas, asimetría en la luz y normalidad en el sonido. QQ de CO2 (S5, S6) mostraron casi normalidad, CO2 Slope (S5) fue normal, y QQ de PIR (S6 y S7) indicaron asimetría en forma de "S". En el boxplot de la luz (S1-S4), distribución no normal y bimodal. No se hallaron valores atípicos. El sonido parece normal, con medianas alrededor de 0 y algunos valores atípicos en S1_Sound y S4_Sound. CO2 (S5_CO2) muestra normalidad, con mediana cerca de 500 ppm. La pendiente de CO2 (S5_CO2_Slope) está sesgada hacia la derecha. Las lecturas de PIR (S6_PIR y S7_PIR) no son normales y son bimodales, con valores atípicos en ambos conjuntos.

Análisis de Normalidad:D'Agostino's K^2 Test p-value en casi todos los casos fue 0.0, rechazando la normalidad al 5%. Anderson-Darling Test Statistic fue significativamente mayor que los valores críticos, concluyendo con confianza que ninguna variable sigue una distribución normal. Temperatura en sensores (S1, S2, S3 y S4): Aproximadamente normal, mediana en 25.75 °C, con ligera asimetría positiva evidente en los bigotes y sin valores atípicos detectados.

Variance Inflation Factor VIF

El VIF es una medida que cuantifica cuánto aumenta la varianza de un coeficiente de regresión debido a la multicolinealidad en el modelo. En términos simples, el VIF ayuda a identificar la colinealidad entre las variables predictoras. Es por eso que aquellas que obtuvieron un resultado por encima de 5 se decidieron no utilizar para el modelo.

Balance de los datos

Se notó una gran diferencia en los datos: 0:8228, 2:748, 3:694, 1:459. Para abordar el desequilibrio, se realizó un remuestreo (upsampling) de la clase minoritaria. El método resample se utilizó para aumentar el número de muestras de la clase minoritaria hasta alcanzar el doble del tamaño de la clase mayoritaria (n_samples=2 * len(majority_data)). Esto se hizo para igualar los recuentos y hacer que las clases estén más balanceadas, quedando como 0:8228, 2:6468, 3:5946, 1:4042.

SGDClassifier

Uno de los algoritmos utilizados fue el Clasificador de Descenso de Gradiente Estocástico, mejor conocido como SGDClassifier, el cual utiliza una técnica de optimización para minimizar una función de pérdida y ajustar los pesos del modelo.

1. Preparación de datos, es decir, separar los datos en un conjunto de test y uno de train esto se realizó por medio de la función train_test_split de scikit-learn.
2. Creación de la pipeline la cual tiene dos etapas: escalado de características utilizando StandardScaler() y clasificación utilizando SGDClassifier. El parámetro loss='log_loss' específica que se utilizará la pérdida de regresión logística para entrenar el clasificador.
3. Entrenar el Modelo: esto se realizó por medio del método fit() con los conjuntos de entrenamiento. Esto ajusta el escalador y el clasificador a los datos de entrenamiento.
4. Predicción y evaluación: Se realizan predicciones en el conjunto de validación utilizando el método predict del pipeline. Ahora bien, también se evaluó el rendimiento del clasificador.
5. Cross Validación: esto se realizó para evaluar el rendimiento del modelo mediante la división del conjunto de datos en varias submuestras y se calcula la precisión promedio de las predicciones.

KNeighborsClassifier

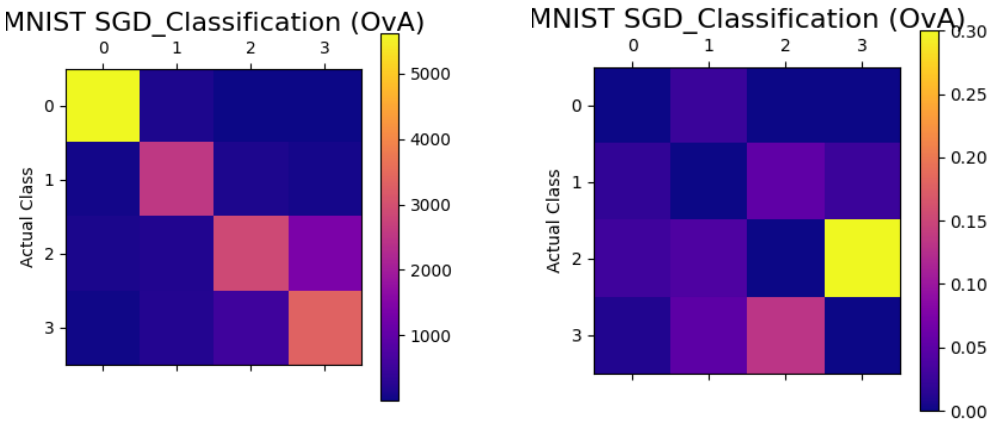
Es un algoritmo de clasificación basado en vecinos más cercanos, es decir, el objetivo es asignar una categoría a un nuevo punto de datos basándose en la mayoría de las etiquetas de clase de sus vecinos más cercanos.

1. Preprocesamiento de Datos y Separación: Se separan las variables independientes y la variable dependiente del conjunto de datos. Luego los datos se separan en test y train utilizando train_test_split. También se realiza un preprocesador con numeric_tranformer que aplica una imputación de la mediana y estandarización a las características numéricas.
2. Pipeline del modelo: luego se crea una pipeline que incluye el preprocesador y para el clasificador se le colocan 15 vecinos.
3. Entrenamiento: Se entrena el modelo a través de la pipeline y se realizan las predicciones.
4. Evaluación: Luego se evaluó el rendimiento del modelo utilizando métricas como precisión, recall, F1, matriz de confusión, r-cuadrado, r-cuadrado ajustado y cross validation.

Resultados

SGDClassifier

Los resultados de la evaluación del modelo son positivos, con una matriz de confusión que revela una alta precisión, especialmente para la Clase 0. Sin embargo, se observa una disminución en la precisión y recall para las otras clases, indicando que el modelo podría tener desafíos al identificar específicamente las clases 2 y 3. Como se puede observar en la matriz siguiente:



El informe de clasificación proporciona detalles adicionales sobre la precisión, recall y F1-score para cada clase, estando todas alrededor de 80-84%. La exactitud global del modelo es del 83.22%. El MSE fue de 0.26, por lo tanto, entre más bajo indica un mejor ajuste del modelo a los datos, ya que implica que las predicciones están más cercanas a los valores reales.

Ahora bien, el coeficiente de determinación (R^2), que evalúa la capacidad del modelo para ajustarse a los datos, estuvo alrededor de 81.07%. Mientras que el R^2 ajustado también se calcula para considerar el número de variables y las instancias en el conjunto de datos, estando en 81.04%. Estos valores indican una adecuada capacidad del modelo para explicar y predecir las clases, aunque hay margen para mejorar especialmente en la identificación de las clases minoritarias. Por otra parte, se realizó una validación cruzada para evaluar el modelo en conjuntos de datos diferentes, se tomaron 5 particiones, lo cual dio como resultado un rango de 83-84%.

KNeighborsClassifier

Los resultados de la evaluación para este algoritmo están cercanas a 100. Técnicamente, esto indicaría un modelo ideal, sin embargo, cabe destacar que esto puede ser causado por overfitting lo cual no es bueno porque indica que nuestro modelo no será bueno para predecir con valores nuevos. Sin embargo, se notó que para este algoritmo también hubo una deficiencia para clasificar las clases 2 y 3. El MSE fue de 0.03, lo que indica que las predicciones están cerca de los valores reales. Por otro lado, para el R^2 y el R^2 ajustado obtuvieron un resultado de 97.78%, lo cual indica que hay una gran porción de la variabilidad de room occupancy que es explicada por las variables independientes. También se realizó un Cross Validation en donde se puede ver que las puntuaciones varían entre aproximadamente 86.94% y 98.28%. Un conjunto de puntuaciones más consistente y cercano entre sí sugiere una mayor estabilidad en el rendimiento del modelo.

	KNeighborsClassifier	**SGDClassifier**
Precisión	98.09%	82.97%
Recall	98.06%	83.22%

f1-score	98.0%	82.97%
Accuracy	98%	83.22%
R^2 Ajustado	97.78%	81.04%

Conclusión

Al abordar el propósito de este informe, que es estimar con precisión el número de ocupantes en una habitación utilizando datos de sensores medioambientales, se han explorado detalladamente diversas facetas del conjunto de datos. La investigación se centró en variables clave como temperatura, luz, sonido, CO2 y PIR, con el objetivo de identificar patrones de ocupación y construir modelos de clasificación efectivos. En cuanto a los resultados, se evidenció la presencia de distribuciones no normales en varias variables, destacando la asimetría y bimodalidad en la luz y los sensores PIR. Se emplearon dos modelos de clasificación, SGDClassifier y KNeighborsClassifier, para prever la ocupación de la habitación. El SGDClassifier presentó resultados sólidos, con una precisión del 83.22%, aunque se identificaron desafíos en la clasificación de clases minoritarias. Por otro lado, el KNeighborsClassifier mostró un rendimiento excepcional, aunque se señaló la posibilidad de *overfitting*. En resumen, este estudio proporciona una visión detallada de la estimación de ocupación basada en datos de sensores. Para futuras investigaciones, se recomienda explorar en mayor profundidad la clasificación de clases minoritarias y gestionar el riesgo de overfitting, posiblemente mediante la recopilación de más datos que abarquen una variedad más amplia de situaciones y contextos. Estos datos adicionales podrían fortalecer la capacidad de los modelos para generalizar y mejorar su rendimiento en condiciones diversas.

Anexo

- Singh, A., Jain, V., Chaudhari, S., Kraemer, F., Werner, S., & Garg, V. (2018). *Machine Learning-Based occupancy Estimation using multivariate sensor nodes.* https://www.semanticscholar.org/paper/Machine-Learning-Based-Occupancy-Estimation-Using-Singh-Jain/e631ea26f0fd88541f42b4e049d63d6b52d6d3ac
- *sklearn.neighbors.KNeighborsClassifier*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- *sklearn.linear_model.SGDClassifier*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

Gráficas Adicionales: