Predicción de condiciones óptimas para generación solar mediante modelos de ML

Proyecto final del Diplomado en Ciencia de Datos con Python
Alumno: Carlos David Leal Fulgencio

Introducción

Se tiene acceso a una base de datos de una estación solarimétrica localizada en el Instituto de Energías Renovables de la UNAM mediante este enlace http://esolmetdata.ier.unam.mx/Esolmet2/index.html#. Los datos corresponden a una marca temporal, la irradiancia directa, la irradiancia global, índice UV, índice UVA, temperatura del aire, humedad relativa, velocidad y dirección del viento, presión barométrica y lluvia. Estos datos se capturan cada 10 minutos, por lo que para un periodo de 13 meses, contamos con más de 56 mil mediciones. Los datos en su plataforma original se ven como lo muestra la Figura 1.

Las estaciones solarimétricas son relevantes porque permiten obtener información sobre el recurso solar en un determinado lugar. Con esta información se puede determinar cómo se comportarían sistemas fotovoltaicos o fototérmicos que aprovechan la energía solar para producir energía eléctrica o térmica.

Obtener los datos presentó el reto de tener que descargar la información en paquetes de 7 días, ya que es lo único que permite la plataforma mediante la que se accede a ellos.

2025-06-29

Rango seleccionado: 6/7/2025, 23:00:00 al 29/6/2025, 23:00:00

Timestamp	I_dir [W/m2]	I_glo[W/m2]	I_dif[W/m2]	Iglo_inc[W/m2]	Indice_UV[W/m2]	Indice_UVA[W/m2]	AirTC[°C]	HR[%]	WS[m/s]	WD[grados]	PB[mbar]	Rain[mm]
5/7/2025, 9:00:00	710.00	531.60	437.00	474.70	0.06	32.14	26.39	55.68	0.33	351.10	867.60	0.00
5/7/2025, 8:50:00	700.10	497.00	407.30	440.70	0.06	29.88	25.78	57.45	0.35	122.70	[illegible]	[illegible]
5/7/2025, 8:40:00	674.40	456.40	372.30	402.30	0.05	27.22	24.95	59.14	0.60	77.91	867.54	0.00
5/7/2025, 8:30:00	652.60	418.10	339.40	364.50	0.04	24.72	24.42	61.81	0.65	116.90	867.49	0.00
5/7/2025, 8:20:00	647.60	382.90	313.00	329.00	0.04	22.59	24.66	60.74	0.63	89.30	867.41	0.00
5/7/2025, 8:10:00	633.30	347.20	284.10	293.40	0.03	20.36	24.77	64.36	0.25	88.10	867.32	0.00
5/7/2025, 8:00:00	596.90	307.00	248.10	255.10	0.02	17.94	24.98	66.04	0.06	89.50	867.16	0.00
5/7/2025, 7:50:00	558.90	267.80	212.30	218.30	0.02	15.59	23.68	68.26	0.00	91.00	866.99	0.00
5/7/2025, 7:40:00	519.90	229.60	177.40	183.60	0.01	13.32	22.18	72.66	0.00	80.80	866.92	0.00
5/7/2025, 7:30:00	490.10	194.70	146.00	151.50	0.01	11.25	21.16	76.82	0.00	57.93	866.76	0.00
5/7/2025, 7:20:00	449.80	160.30	115.90	120.70	0.01	9.28	19.54	82.60	0.57	55.42	866.59	0.00
5/7/2025, 7:10:00	399.50	126.80	78.44	91.60	0.01	7.44	18.83	85.10	0.56	345.10	866.37	0.00
5/7/2025, 7:00:00	339.90	95.70	50.55	65.73	0.01	5.75	18.32	87.40	1.23	334.80	866.17	0.00
5/7/2025, 6:50:00	274.70	67.50	41.60	44.03	0.00	4.25	18.28	86.60	1.37	327.40	866.06	0.00
5/7/2025, 6:40:00	190.70	43.17	33.19	31.71	0.00	2.93	17.82	88.80	1.59	0.15	865.90	0.00
5/7/2025, 6:30:00	80.70	23.45	20.01	19.89	0.00	1.82	17.66	89.50	2.13	324.00	865.79	0.00
5/7/2025, 6:20:00	14.94	9.83	8.87	8.37	0.00	0.94	17.67	89.80	1.99	330.40	865.62	0.00
5/7/2025, 6:10:00	0.14	2.62	2.68	1.87	0.00	0.36	17.51	89.30	1.97	327.90	865.49	0.00
5/7/2025, 6:00:00	0.02	0.15	0.08	0.00	0.00	0.08	17.39	91.70	1.85	338.00	865.43	0.00

Figura 1. Datos mostrados por la plataforma web de la estación solarimétrica del IER-UNAM

Problema

Predecir si las condiciones ambientales son óptimas para máxima generación de energía solar (binario: Sí/No). Se considerará una irradiancia global mínima de 600 W/m^2 como parámetro que determine si hay condiciones adecuadas para la generación de energía eléctrica.

Objetivo

Desarrollar un modelo de clasificación que, usando variables meteorológicas, prediga ventanas de alta eficiencia energética.

Variables

- Target: Clase binaria derivada de irradiancia_global (ej: 1 si >600 W/m², 0 si ≤600).
- Features: Todas las demás (temperatura, humedad, viento, etc.).

Impacto

Permitir a plantas solares anticipar periodos de alta producción y ajustar mantenimiento.

Metodología y resultados

Preprocesamiento de datos.

1. Se importan los datos desde GitHub
2. Se eliminaron filas con datos faltantes que se generaron al descargar la información.
3. Se renombraron columnas para un mejor entendimiento y se transformó la columna de «Marca de tiempo» para que los datos pudieran ser tratados con Pandas
4. Una de las columnas tenía información en formato «object», se cambió por «float»
5. Creación de la variable «nubosidad» o «razón_difusa/global»
6. Se elimina la variable de irradiancia global, ya que a partir de esta se creó el *target*. Esto se considera importante para evitar la fuga de datos
7. Comprobación del balance de clases según el *target* (irradiancia global):
    a. 80.85 % para «0» o para baja irradiancia global (<600 W/m^2)
    b. 19.15 % para «1» o para alta irradiancia global (>600 W/m^2)

División del dataframe y escalamiento de variables

1. Se usó la función *train_test_split* de *scikit-learn* para dividir los datos en conjuntos de entrenamiento, validación y prueba.
2. Se usó el argumento *stratify* para mantener la proporción de clases en el conjunto de entrenamiento y prueba. Las dimensiones de estos conjuntos son, respectivamente: (38664, 14), (9667, 14) y (8529, 14).

3. Se calcularon los pesos de las clases para el conjunto de entrenamiento, algo muy útil en este caso en que las clases de la variable *target* presentan desbalance. Los pesos encontrados fueron de:
    a. {0: (0.6184458875843757), 1: (2.6106684672518568)}
4. Se realizó el escalamiento mediante la función *MinMaxScaler* de *Scikit-Learn*.

Creación del modelo

1. Se definió una función para la creación de un modelo de red neuronal que permite elegir entre dos opciones (o más si se edita) que difieren por la cantidad de capas y neuronas.
2. Se usó activación *relu* para las capas densas; se usó activación *sigmoid* para la capa de salida con una neurona (clasificación binaria), y para compilar se usó optimización *Adam* con la función de pérdida *binary cross-entropy* y como métrica para evaluar el desempeño del modelo el *recall*.
3. Definimos un parámetro de paro temprano que monitorea el *val_loss* (pérdida de validación) y restaura los pesos de las clases a aquellos que se calcularon en la época en que conseguimos la mejor métrica monitoreada.
4. Se definió el entrenador que incorpora la parada temprana y los pesos que calculamos anteriormente.

Desempeño del modelo

Se definió una celda cuyo código permite visualizar el desempeño del modelo (Figura 2). Se observa que el desempeño fue bueno, principalmente al imprimir los elementos de evaluación:

Accuracy: 0.9669363348575448
Recall: 0.9840783833435395
Precision: 0.8625872249060654
F1 Score: 0.9193363844393593

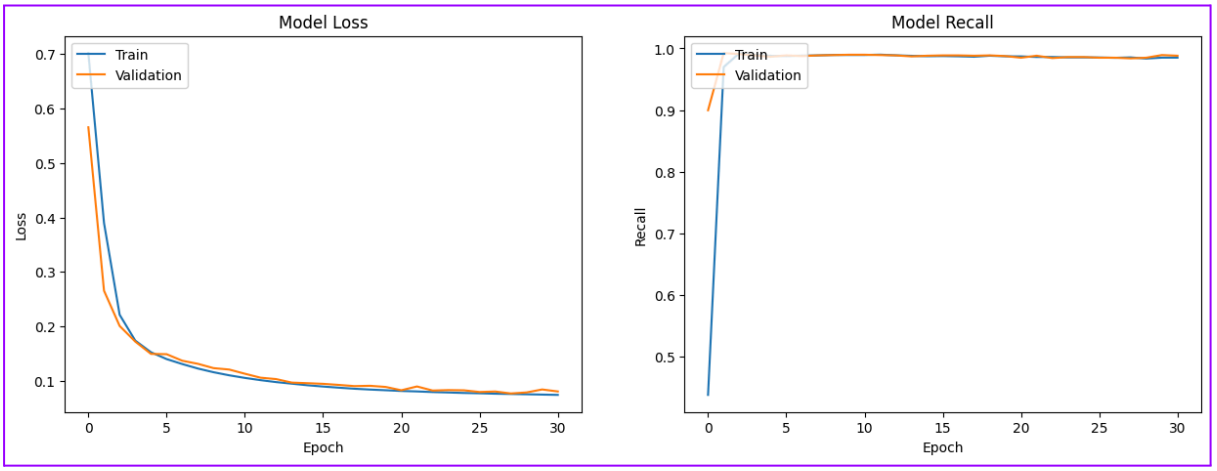

Figura 2. Visualización del historial del entrenamiento del modelo de redes neuronales

Como parte de la visualización del desempeño, imprimimos también la matriz de confusión mediante las etiquetas verdaderas (y_test) y las predichas (y_pred). La matriz de confusión es muy útil para ver el desempeño del modelo de clasificación, especialmente en casos con datasets desbalanceados, como es este. De acuerdo con la Figura 3, podemos

ver que el modelo se comporta bien; sin embargo, se cometen algunos errores al detectar falsos positivos (alrededor del 14 %).

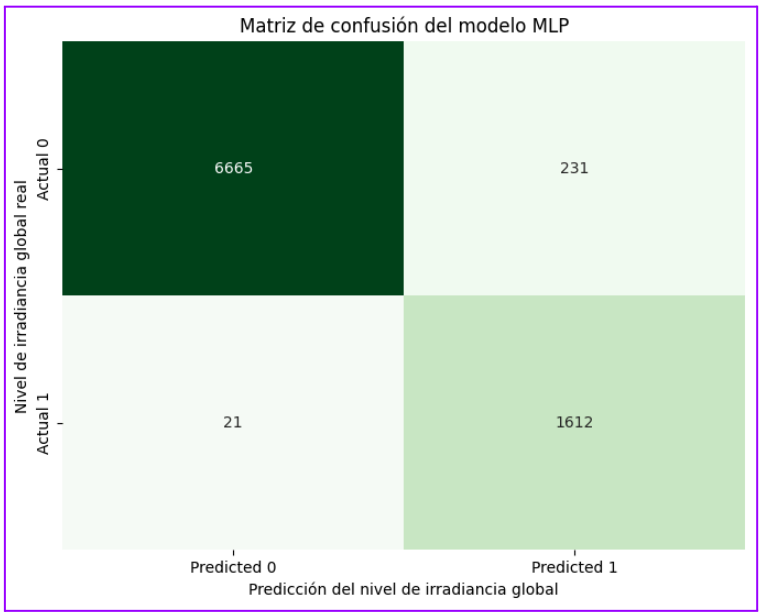

Figura 3. Matriz de confusión para las predicciones del modelo en el conjunto de prueba.

Modelo Random Forest

Dado que tenemos una precisión relativamente baja según lo calculado arriba (91 %), se decidió implementar otro modelo de clasificación que no implique redes neuronales. Se optó por un modelo Random Forest, por ser una opción robusta y versátil.

Después de definir una *Grid Search* para definir los mejores hiperparámetros, se generó un modelo *Random Forest* con *n_stimators=100*, *max_depth=None*, *min_samples_split=5*, y *class_weigth='balanced'*, para lidiar con el desbalance de clases de este problema, sin necesidad de hacer un balanceo de clases específico con alguna otra función.

Este modelo de *Random Forest* tuvo un desempeño muy bueno, como podemos ver:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	13791
1	0.96	0.99	0.98	3267
accuracy			0.99	17058
macro avg	0.98	0.99	0.98	17058
weighted avg	0.99	0.99	0.99	17058

Estos resultados hacen pensar un posible problema de fuga de datos; sin embargo, no fue posible determinar si esto es así.

En la Figura 4 se muestra un gráfico que permite inspeccionar visualmente la precisión con la que las predicciones del modelo *Random Forest* se alinean con las condiciones óptimas reales a lo largo del tiempo. Se puede ver que el modelo predice correctamente los períodos óptimos; por otro lado, prácticamente no se ven errores (falsos positivos o falsos negativos), porque no se presentan casos donde no se superponen los puntos "Real" y "Predicho".

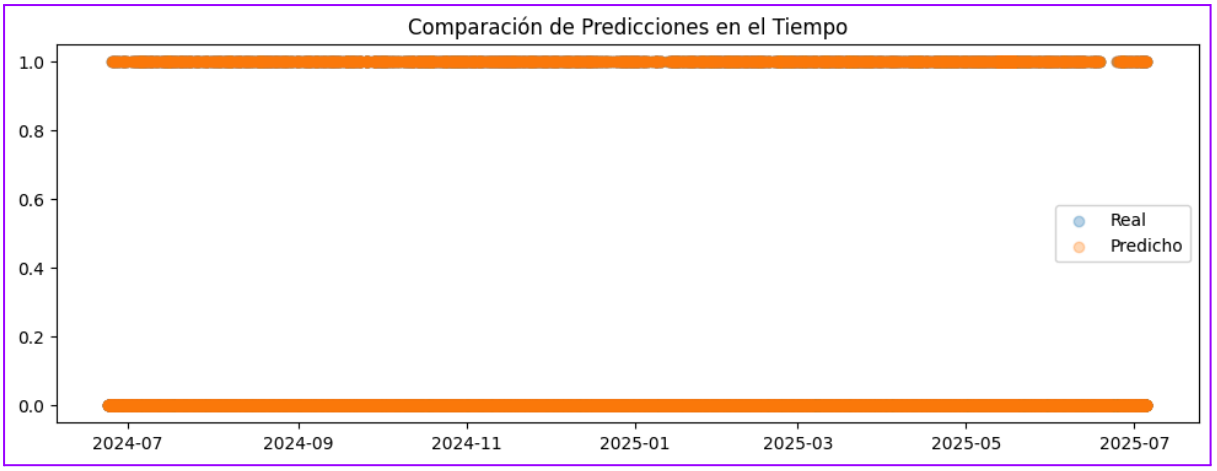

Figura 4. Diagrama de dispersión para condiciones óptimas reales y previstas para la generación solar.

Por último, visualicemos cuáles variables son las más relevantes en la predicción que el modelo Random Forest realiza (Figura 5).

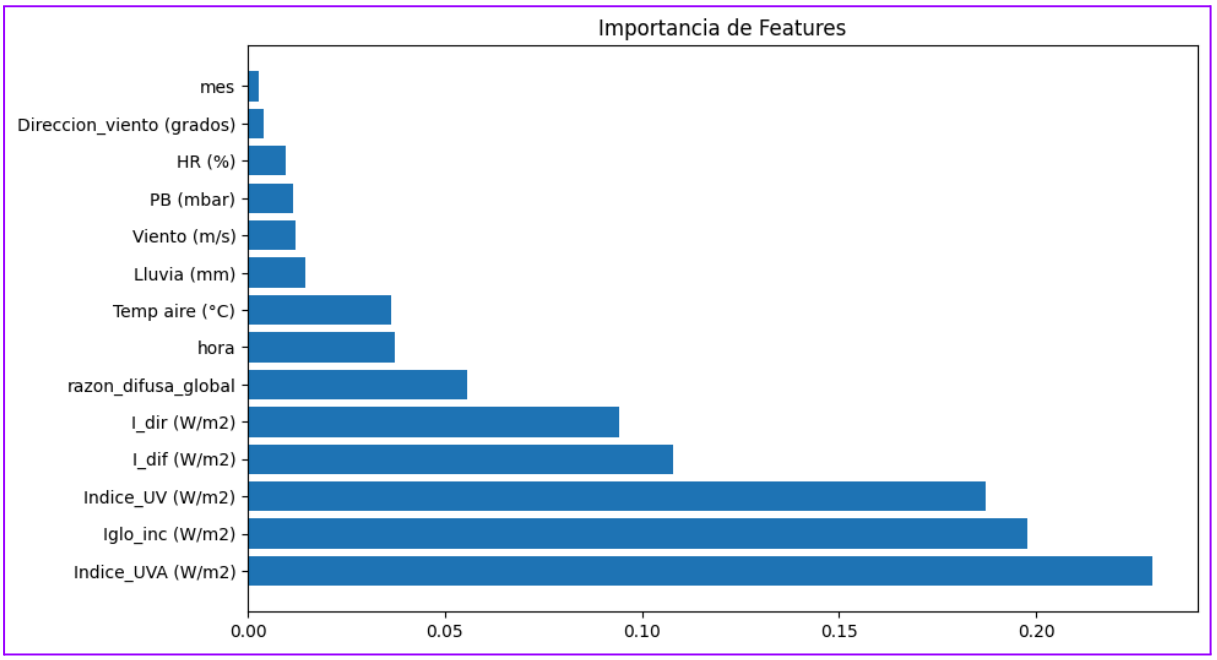

Figura 5. Variables y su relevancia en las predicciones del modelo de Random Forest.

Conclusiones

1. Ambos modelos implementados, clasificación binaria usando redes neuronales y clasificación usando Random Forest, son útiles para predecir ventanas de alta generación de energía, por lo que es posible planificar mantenimiento en periodos de baja producción.
2. El modelo de Random Forest tuvo un desempeño mejor que el de redes neuronales. De hecho, tuvo tan buen rendimiento que hace pensar que hubo fuga de datos; sin embargo, no fue posible identificarla.

3. Variables como los índices UV y UVA son críticas para hacer predicciones de alta irradiancia global o condiciones ideales para generación de energía.
4. La variable «nubosidad» o «razón_difusa/global» no fue relevante como se esperaba.
5. Es posible que el excelente desempeño que tuvo el modelo de Random Forest se deba a que las otras variables (las que no son irradiancia global ni los índices de radiación UV) tienen muy poca influencia y el problema planteado es, hasta cierto punto, ocioso.

Siguientes pasos sugeridos

Pueden implementarse otras estrategias, como postprocesamiento, y eliminar algunas de las variables que podrían parecer más ovias.

Algo interesante que puede hacerse a continuación es la segmentación mediante clusterización para predecir condiciones para el mantenimiento de módulos fotovoltaicos. Los pasos para realizar esto podrían ser los siguientes:

1. Selección de datos para la clusterización. Decidir qué características del *DataFrame* son las más significativas para identificar patrones climáticos relacionados con el bajo rendimiento de los paneles solares. Según las conclusiones planteadas, se podría excluir características como «razon_difusa/global», pero incluir otras como la irradiancia global, los índices UV, la temperatura, la humedad, etc.
2. Preprocesamiento para la clusterización. Probablemente, será necesario escalar las características seleccionadas utilizando técnicas como *StandardScaler* o *MinMaxScaler*.
3. Elección de un algoritmo de clusterización adecuado. K-Means es una opción común, pero también puede considerar otras opciones.
4. Determinación del número óptimo de agrupaciones. Con un algoritmo como K-Means, habrá que determinar el número óptimo de agrupaciones. El método del codo o la puntuación de silueta son técnicas populares para esto.
5. Implementación del algoritmo de agrupación. Aplicar el algoritmo de agrupación elegido a los datos preprocesados.
6. Análisis e interpretación de las agrupaciones. Analizar las características de cada agrupación para comprender las condiciones climáticas que representan. Examinar la estadística (media o mediana) de las características dentro de cada agrupación parece lo más prudente.
7. Relacionar las agrupaciones con bajo rendimiento. Conectar las agrupaciones identificadas de patrones climáticos con períodos de bajo rendimiento de los paneles solares. Se podría usar el mismo umbral (600 W/m^2) para identificar «bajo rendimiento» u otras métricas relevantes.
8. Visualizar las agrupaciones. Esto podría implicar diagramas de dispersión de puntos de datos coloreados según su asignación de clúster u otras visualizaciones que ayuden a comprender la estructura del clúster.
9. Desarrollo de recomendaciones de mantenimiento. Formulación de recomendaciones para acciones de mantenimiento predictivo que se deben tomar cuando se detecten patrones climáticos específicos (clústeres).