Predicción del clima

Proyecto final | Coder House | Data Science

Autor: Alexander Daniel Rios

Tabla de contenido

1. Descripción del caso de negocio

A menudo pasamos desapercibido la importancia del pronóstico, pero muchas veces nos vemos obligados a realizar cambios en nuestra rutina diaria de acuerdo a este.

Es por esto que existen múltiples organizaciones alrededor del mundo que brinda y analizan datos de determinados factores meteorológicos en rigor de poder predecir eventos climáticos que puedan afectar la vida de las personas. Open Meteo es una API meteorológica que nos brinda datos tales como la temperatura, presión, velocidad del viento, etc. de diversas partes del mundo.

2. Objetivos del proyecto

Nuestro objetivo a través de este proyecto será analizar una serie de datos en busca de aquellas variables más influyentes para la predicción de precipitaciones, esto será llevado adelante a través del análisis exploratorio de los datos, visualizaciones, modelos predictivos y métricas de comparación.

Buscaremos responder las siguientes preguntas claves:

- ¿Existira algun patron que nos permita identificar un dia posiblemente lluvioso?
- ¿Qué parámetros son vitales para predecir un dia lluvioso? Y de ser asi, ¿Como se relacionan?
- ¿Habrá una época del año en la que haya mayores probabilidades de llover?

3. Descripción de los datos

Parámetros horarios

Variable	Descripción
time	Fecha y hora de referencia
temperature	Temperatura del aire
relativehumidity	Humedad relativa
dewpoint	Temperatura de rocío
apparent_temperature	Sensación térmica
surface_pressure	Presión atmosférica del aire
precipitation	Suma de precipitación total de la hora anterior
rain	Precipitación líquida
snowfall	Cantidad de nieve precipitada
cloudcover	Cobertura total de nubes como fracción de área
shortwave_radiation	Radiación solar de onda corta
direct_radiation	Radiación solar directa
diffuse_radiation	Radiación solar difusa
windspeed	Velocidad del viento
winddirection	Dirección del viento
windgusts	Refagas de viento
evapotranspiration	Evapotranspiración
weathercode	Condición climática como código numérico
snow_depth	Profundidad de la nieve en el suelo
vapor_pressure	Falla de presión de vapor
soil_temperature	Temperatura promedio del suelo bajo tierra
soil_moisture	Contenido medio de agua del suelo

Parámetros diarios

Variable	Descripción
time	Fecha y hora de referencia
weathercode	Condición climática como código numérico
temperature (max and min)	Temperatura diaria máxima y mínima del aire
apparent_temperature (max and min)	Sensación térmica diaria máxima y mínima
precipitation_sum	Suma de la precipitación diaria
rain_sum	Suma de la lluvia diaria
snowfall_sum	Suma de nevada diaria
precipitation_hours	Número de horas con lluvia
sunrise and sunset	Horas de salida y puesta del sol
windspeed and windgusts	Velocidad máxima del viento y ráfagas en un día
winddirection	Dirección dominante del viento
shortwave_radiation_sum	La suma de la radiación solar en un día
evapotranspiration	Suma diaria de evapotranspiración

La API nos proporciona dos posibles set de datos, datos diarios y datos horarios, ambos set de datos han sido utilizados. Esto último fue posible debido a que tenemos la variable ‘time’ que nos determina la referencia temporal en la cual fueron relevados los registros y mediante la cual fue posible combinar ambos set sin perder la secuencia temporal que veremos que es vital para llevar adelante las predicciones.

Tipos de datos

Veremos a continuación que la mayoría de las variables son numéricas, pero no obstante nos encontraremos con variables del tipo ‘date’, las cuales debemos codificar para poder analizarlas y procesarlas.

Datos horarios

#	Column	Non-Null Count	Dtype
0	time	210936 non-null	object
1	temperature_2m	210909 non-null	float64
2	relativehumidity_2m	210909 non-null	float64
3	dewpoint_2m	210909 non-null	float64
4	apparent_temperature	210909 non-null	float64
5	precipitation	210909 non-null	float64
6	rain	210909 non-null	float64
7	snowfall	210909 non-null	float64
8	weathercode	210909 non-null	float64
9	pressure_msl	210909 non-null	float64
10	surface_pressure	210909 non-null	float64
11	cloudcover	210909 non-null	float64
12	cloudcover_low	210909 non-null	float64
13	cloudcover_mid	210909 non-null	float64
14	cloudcover_high	210909 non-null	float64
15	et0_fao_evapotranspiration	210909 non-null	float64
16	vapor_pressure_deficit	210909 non-null	float64
17	windspeed_10m	210909 non-null	float64
18	windspeed_100m	210909 non-null	float64
19	winddirection_10m	210909 non-null	float64
20	winddirection_100m	210909 non-null	float64
21	windgusts_10m	210909 non-null	float64
22	soil_temperature_0_to_7cm	210909 non-null	float64
23	soil_temperature_7_to_28cm	210909 non-null	float64
24	soil_temperature_28_to_100cm	210909 non-null	float64
25	soil_temperature_100_to_255cm	210909 non-null	float64
26	soil_moisture_0_to_7cm	210909 non-null	float64
27	soil_moisture_7_to_28cm	210909 non-null	float64
28	soil_moisture_28_to_100cm	210909 non-null	float64
29	soil_moisture_100_to_255cm	210909 non-null	float64
30	is_day	210936 non-null	int64
31	shortwave_radiation	210909 non-null	float64
32	direct_radiation	210909 non-null	float64
33	diffuse_radiation	210909 non-null	float64
34	direct_normal_irradiance	210909 non-null	float64
35	terrestrial_radiation	210936 non-null	float64
36	shortwave_radiation_instant	210909 non-null	float64
37	direct_radiation_instant	210909 non-null	float64
38	diffuse_radiation_instant	210909 non-null	float64
39	direct_normal_irradiance_instant	210909 non-null	float64
40	terrestrial_radiation_instant	210936 non-null	float64

Datos diarios

#	Column	Non-Null Count	Dtype
0	time	8789 non-null	object
1	weathercode	8788 non-null	float64
2	temperature_2m_max	8788 non-null	float64
3	temperature_2m_min	8788 non-null	float64
4	temperature_2m_mean	8787 non-null	float64
5	apparent_temperature_max	8788 non-null	float64
6	apparent_temperature_min	8788 non-null	float64
7	apparent_temperature_mean	8787 non-null	float64
8	sunrise	8789 non-null	object
9	sunset	8789 non-null	object
10	precipitation_sum	8787 non-null	float64
11	rain_sum	8787 non-null	float64
12	snowfall_sum	8787 non-null	float64
13	precipitation_hours	8789 non-null	float64
14	windspeed_10m_max	8788 non-null	float64
15	windgusts_10m_max	8788 non-null	float64
16	winddirection_10m_dominant	8787 non-null	float64
17	shortwave_radiation_sum	8787 non-null	float64
18	et0_fao_evapotranspiration	8787 non-null	float64

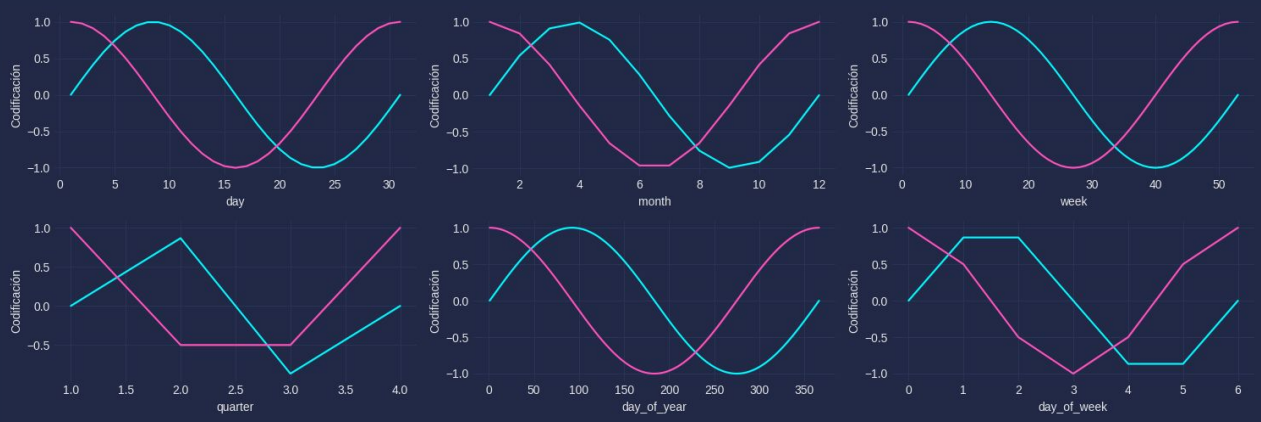
4. Ingeniería de datos

La primera variable sobre la cual debemos trabajar es ‘time’ la fecha de referencia, una fecha contiene más información de la que normalmente tenemos en cuenta.
Una fecha puede ser descompuesta en diversas variables numéricas equivalentes así como lo son el mes, día, año, día de la semana, dia del año, trimestre y semana, a continuación se muestra un ejemplo:

	date	day	month	year	week	quarter	day_of_year	day_of_week
2000-01-01	1	1	2000	52	1	1	5
2000-01-02	2	1	2000	52	1	2	6

Codificaciones ciclicas

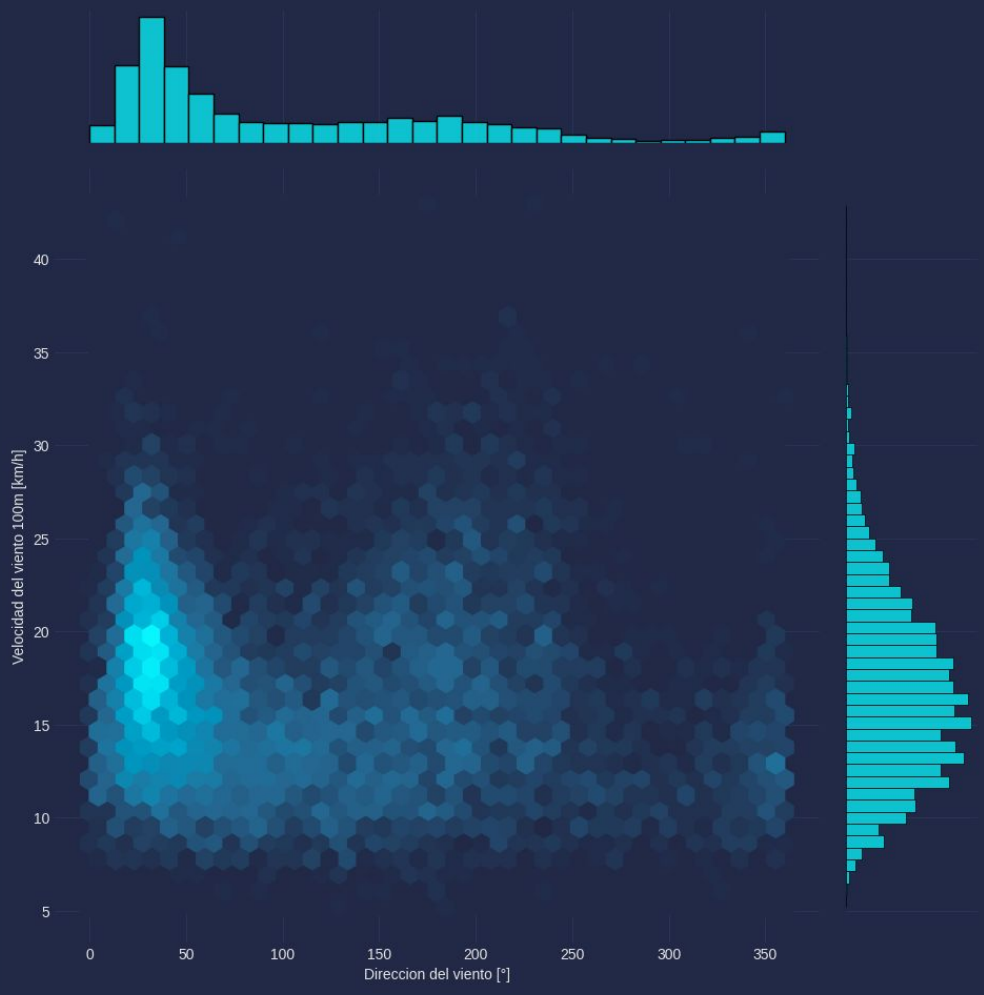
Las funciones periódicas son funciones que se comportan en una manera cíclica sobre un periodo específico, con excepción del año, el resto de variables temporales pueden ser representadas por funciones periódicas (tales como seno o coseno). Por ejemplo, un año siempre tiene 12 meses, por lo que la variable ‘month’ puede ser representada por una función periódica con un periodo de 12 meses. De esta forma podemos codificar las anteriores variables.



Un pre-procesamiento similar al anterior fue realizado con las variables ‘sunset’ y ‘sunrise’.

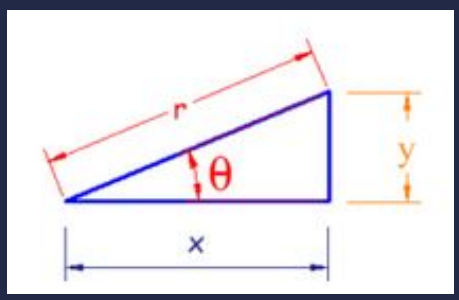
Vector de viento

Entre algunos de los datos disponibles contamos con ‘windspeed’ medida en km/h y ‘winddirection’ medido en grados sexagesimales (°), los cuales pueden describir un vector denominado ‘vector viento’.



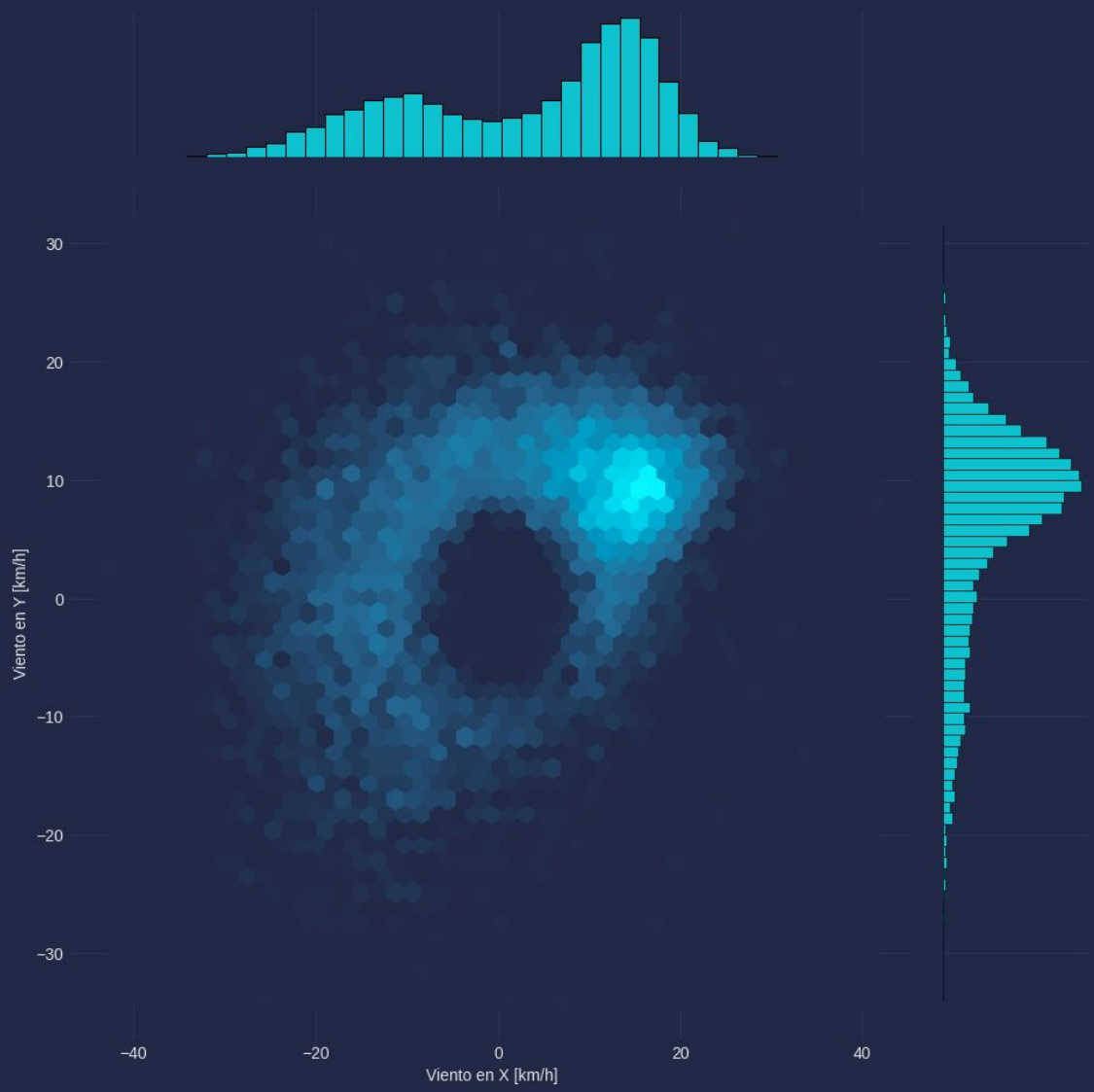
En un sistema polar de coordenadas un vector puede describirse a partir de un ángulo y un radio. De esta forma, podemos considerar:

r = windspeed
θ = winddirection



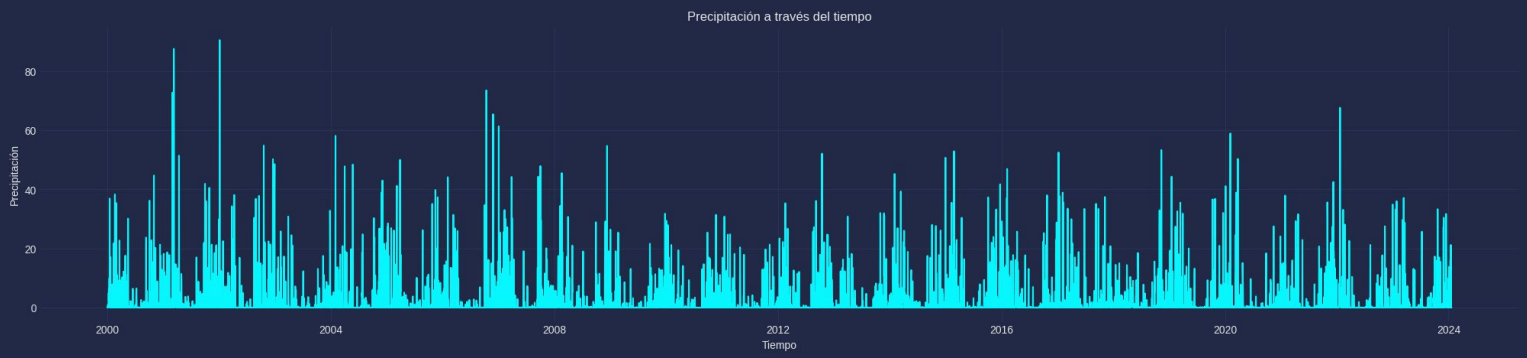
A través de una conversión de coordenadas polares a cartesianas podemos descomponer estas dos variables en otras 2 equivalentes.

Por medio de la siguiente conversión: $x = r \cdot cos(\theta)$ $y = r \cdot sen(\theta)$ obtenemos las componentes ‘wind_x’ y ‘wind_y’ del vector viento.

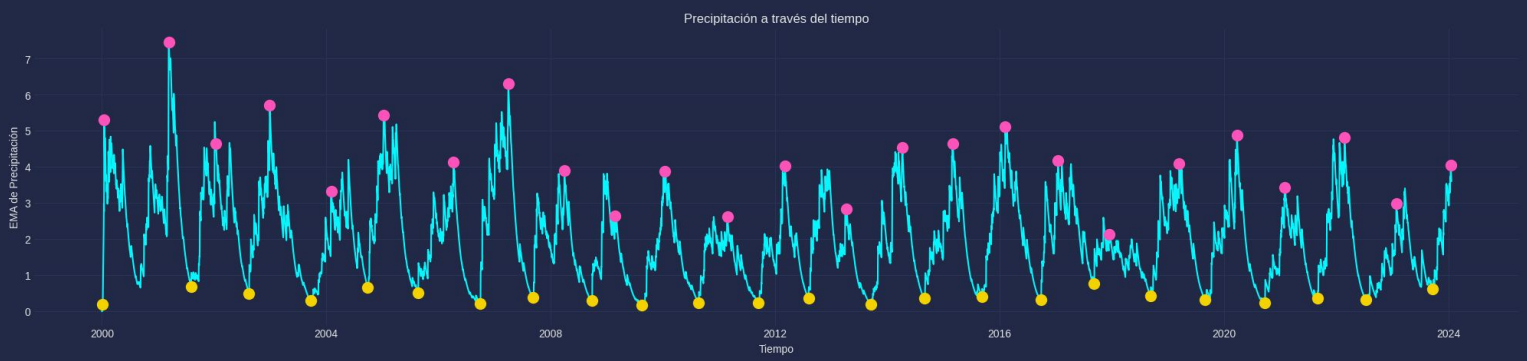


5. Exploratory data analysis

Nuestra variable de interés es ‘precipitation’ ya que es la que buscamos predecir, por lo tanto fue la primera sobre la que comenzamos el análisis. Al ser nuestros datos series temporales la visualización por excelencia son los graficos de lineas.

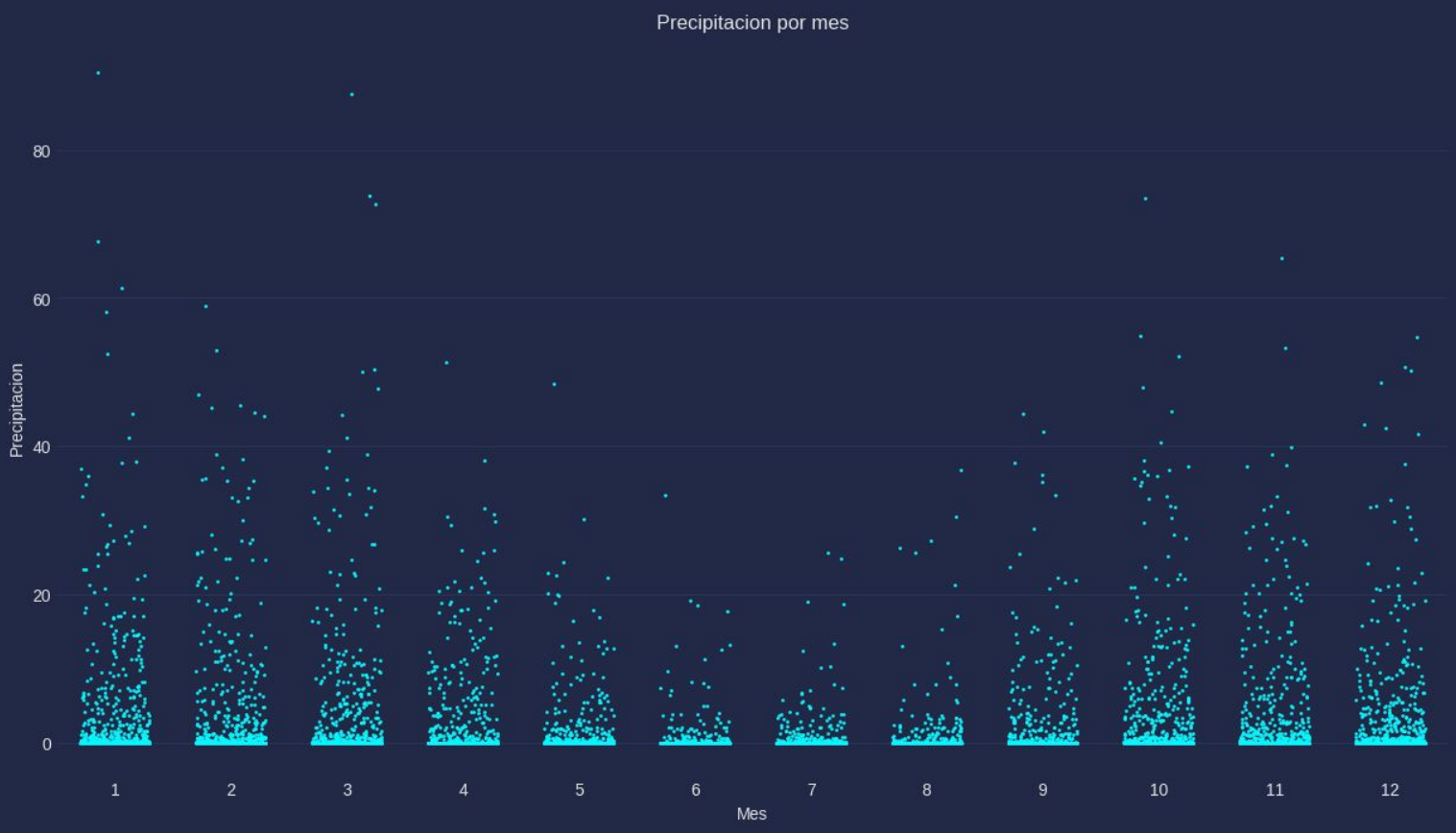


Se ve claramente que las precipitaciones describen un patrón cuasi-periodico, para verlo más claramente se realizó un suavizado denominado Exponential smoothing.



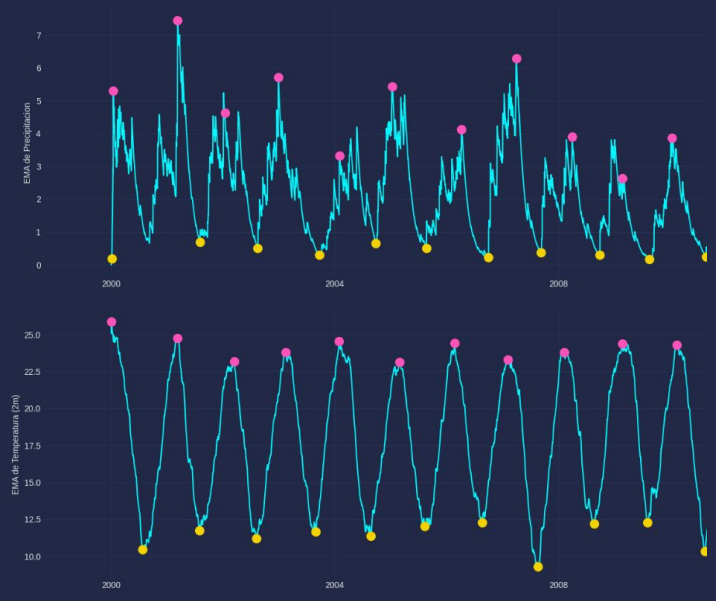
Este procedimiento nos permite observar la periodicidad de esta variable, donde podemos ver que en un periodo de 4 años las precipitaciones alcanzan 4 máximos locales y 4 mínimos locales.

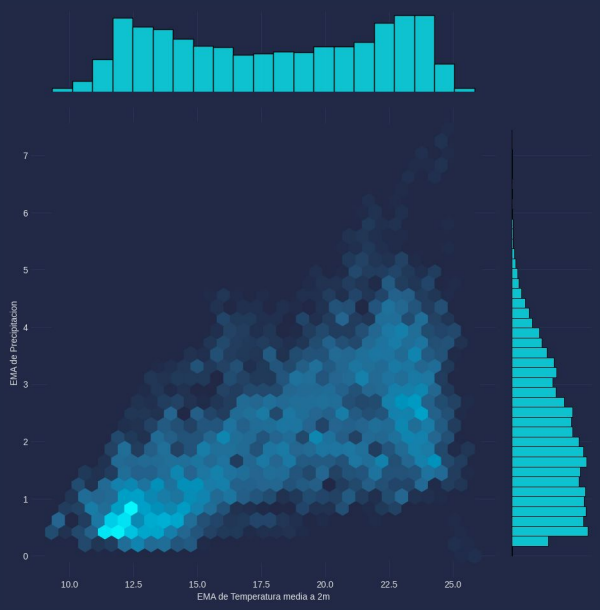
Analizando más a fondo, podemos determinar que un año hay un periodo de acumulacion abundante de precipitaciones y un periodo de acumulacion baja de precipitaciones.



- Periodo de acumulacion abundante de precipitaciones: Diciembre, Enero, Febrero, Marzo y Abril
- Periodo de acumulacion baja de precipitaciones: Julio, Agosto, Septiembre y Enero

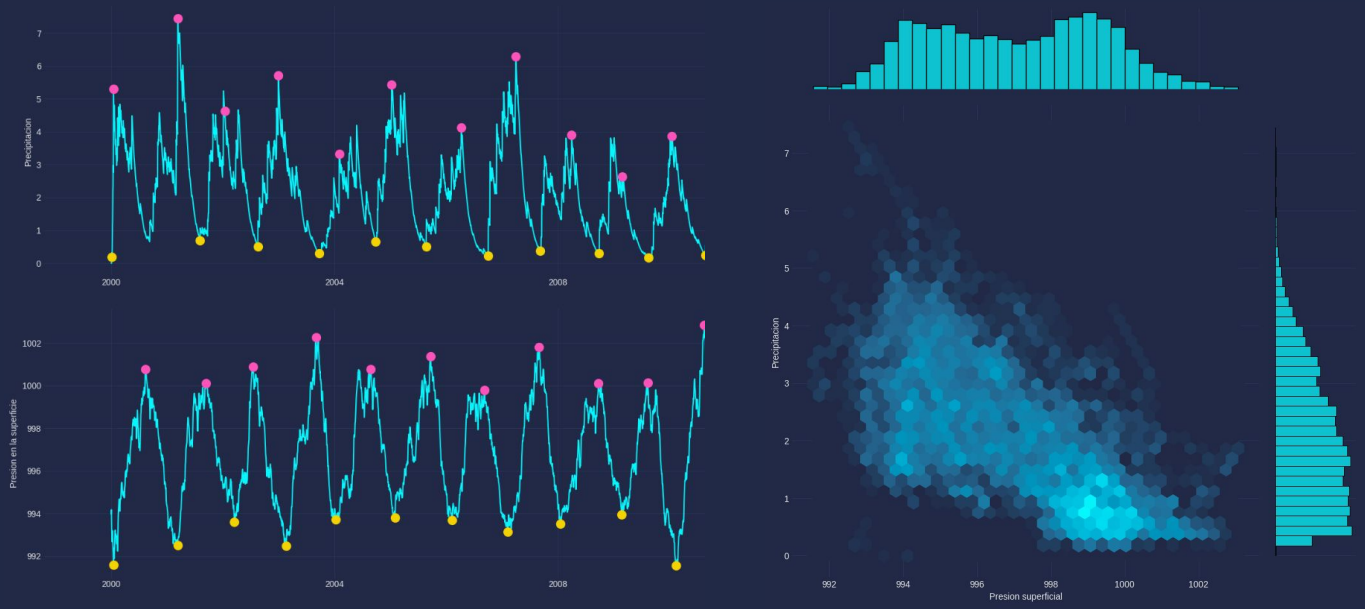
Estas observaciones nos han llevado a pensar si esta periodicidad se repite en otras variables, las precipitaciones son agua que anteriormente se evaporó y se condensó en la atmósfera, una causal de esto puede ser la temperatura. Así que esta fue la siguiente variable de estudio. Para no repetir procesos, al igual que antes suavizamos nuestra variable para detectar periodicidad.





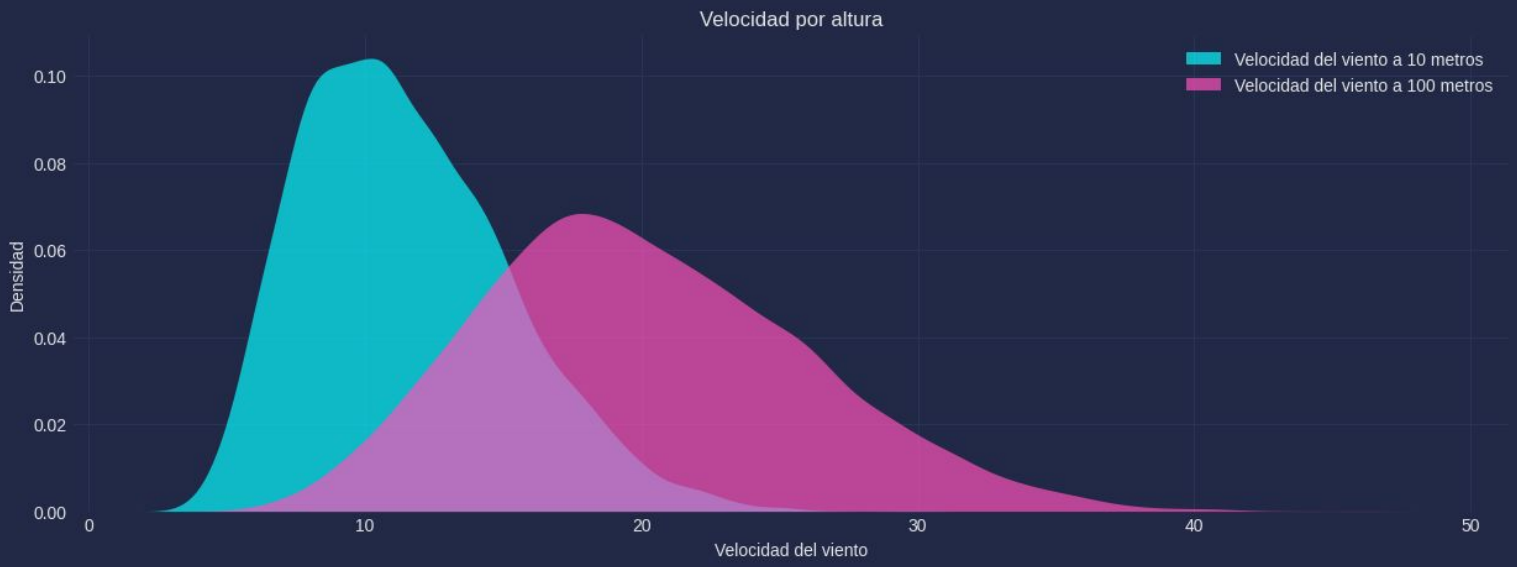
Podemos ver que la temperatura también sigue un patrón periodico, y como podemos ver coincide con la periodicidad en las precipitaciones, de esta forma pudimos comprobar nuestra hipótesis, pues la temperatura juega un papel importante en la evaporación del agua, es por esto que en temporadas de altas temperaturas las precipitaciones son más abundantes y en las temporadas de bajas temperatura las precipitaciones son menos abundantes.

De forma similar procedimos con la presión sobre la superficie.



Pero a diferencia de lo que sucedía con la temperatura, aquí las precipitaciones máximas se dan cuando la presión superficial es mínima, esto es debido a que al haber menor presión sobre una zona, el aire de los alrededores será 'absorbido' con mayor facilidad.
Esto sucede así porque las presiones bajas indican zonas de aire menos concentrado y el aire de los alrededores, que están en zonas de aire mas concentrado (presion alta), tenderá a desplazarse a zonas de aire menos concentrado, arrastrando consigo vapor de agua, facilitando así la formación de nubosidades, y estas últimas que ya no estarán formadas por vapor, si no, por agua se precipitan posteriormente sobre la superficie.

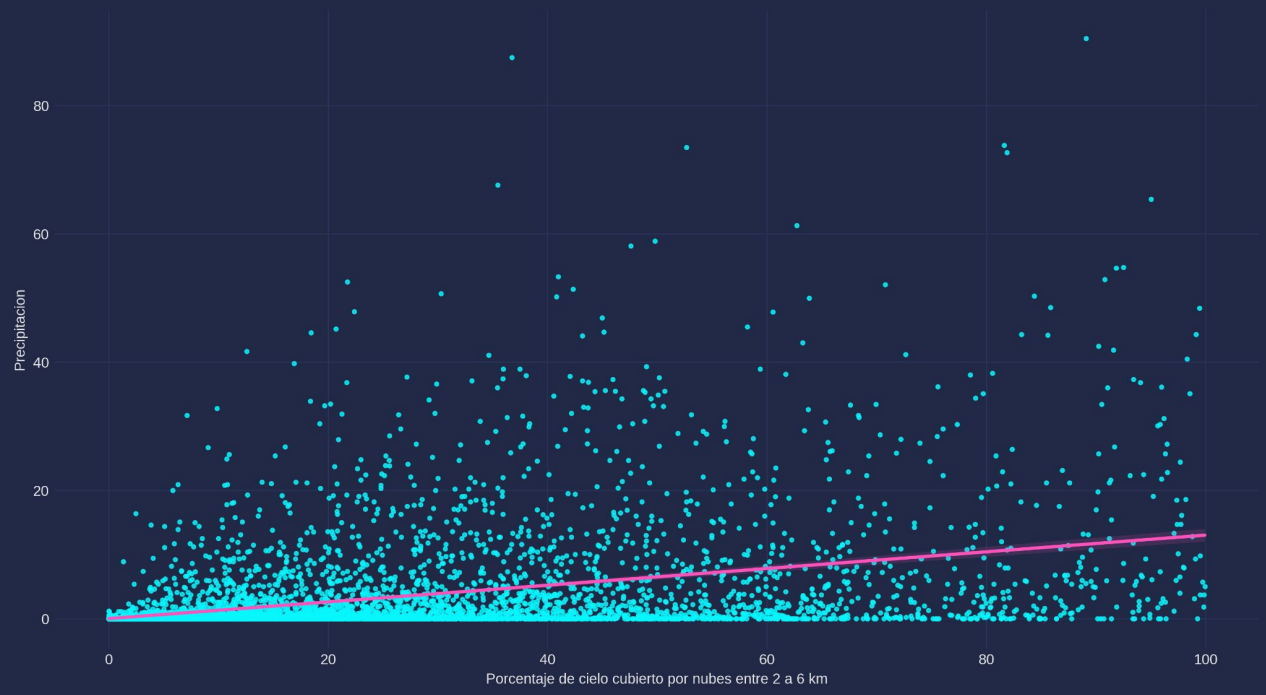
Otro parámetro que considere importante fue el viento, ya que el viento es capaz de desplazar a las nubes a grandes distancias.



	count	mean	std	min	25%	50%	75%	max
windspeed_10m_mean	8787.0	11.452418	3.74334	3.625	8.58125	10.983333	13.887500	29.725000
windspeed_100m_mean	8787.0	20.088114	5.88439	5.375	15.84375	19.529167	24.016667	46.141667

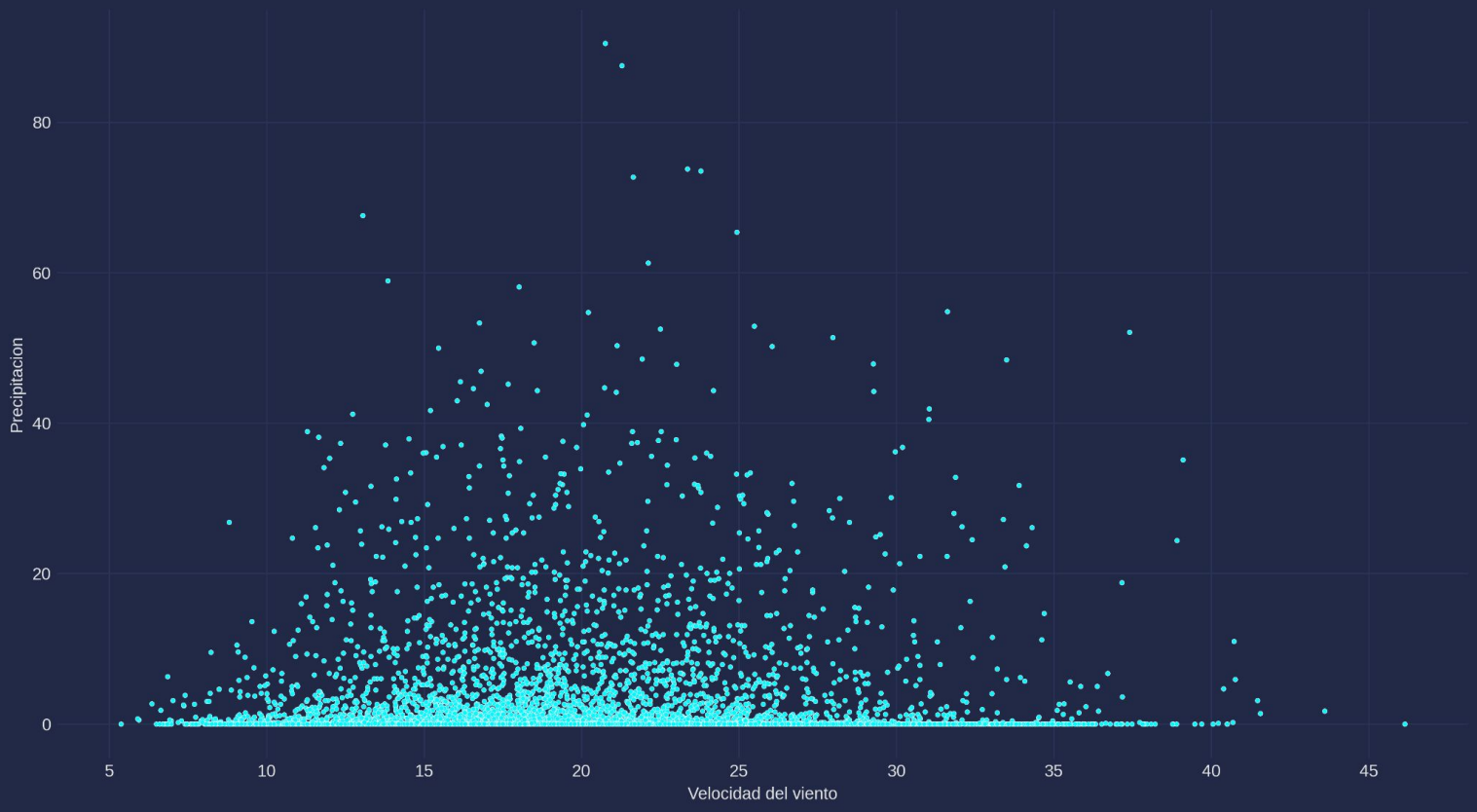
Como podemos ver cuanto mayor es la altura de medición mayor es la velocidad media que podemos detectar, esto puede deberse a múltiples factores, pero quizás la explicación más intuitiva es que el aire a mayor altura es probable que circule con mayor libertad y enfrenta menos obstáculos lo que facilita un incremento de la velocidad.

Y te preguntarás ¿Y esto cómo influye en las precipitaciones?.
Bueno es muy simple, si el aire circula fácilmente sobre la atmósfera desplazando nubosidades, es decir, facilitando la acumulacion de vapores de agua, creara areas de alta concentración de nubosidades que mas temprano que tarde se podrían precipitar.



Como es de esperarse, el porcentaje de cielo cubierto es mayor cuanto mayor son las precipitaciones, pero la tendencia es leve.

Bien, entendemos que el viento juega un rol importante en el desplazamiento de las nubosidades, pero mayor velocidad en el viento no implica mayores precipitaciones.



Es contraintuitivo, pero así como los fuertes vientos pueden arrastrar lluvias también pueden alejarlas, por lo que es lógico pensar que en el momento de mayores precipitaciones la velocidad del viento sea menor una vez que la precipitación está establecida, se esperaría que la velocidad sea mayor hasta arrastrarla dentro del territorio donde se realiza el sondeo, luego debería comenzar a disminuir.

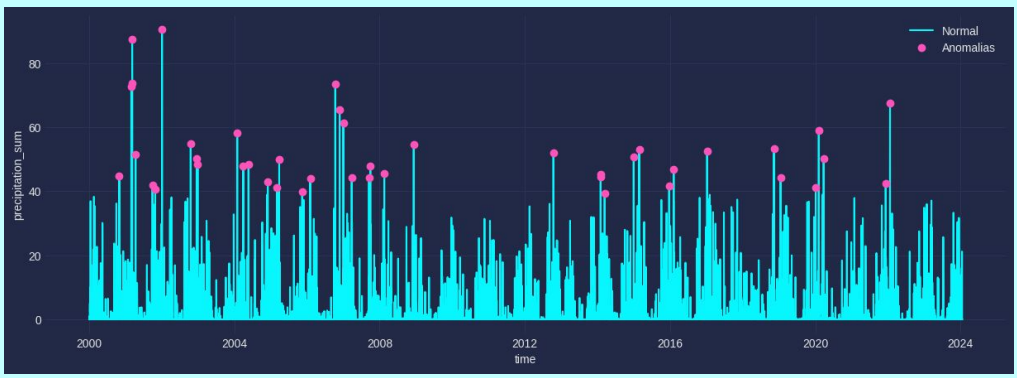
Detección de outliers

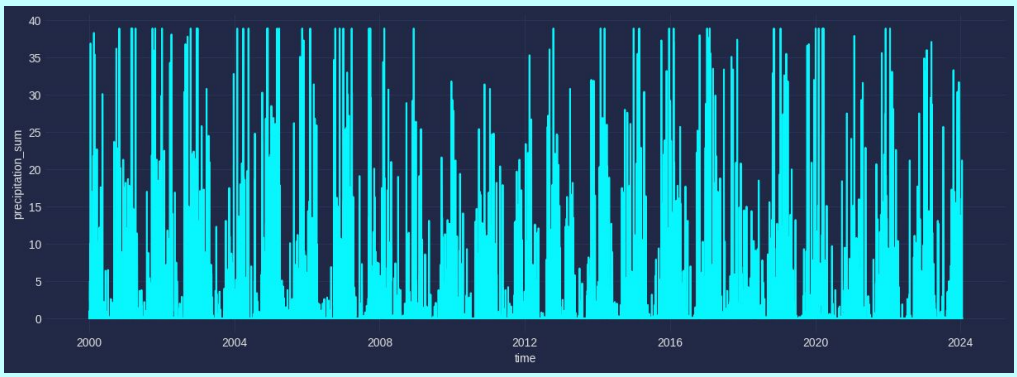
Se analizaron todas las variables y se determinó que la mayoría de los datos se encuentran en rango normales. Sin embargo, debemos destacar que nuestra variable objetivo será también un dato de entrada y seguramente el más importante de todos, ya que nuestra tarea principal es realizar un ‘forecasting’, es decir predecir el valor futuro de una variable determinada. Es por esto que intentaremos suavizar la variable “precipitation_sum” de forma tal de acotar el rango dinámico de la variable.

Detección con árboles de clasificación (CART)

Podemos utilizar el poder y la solidez de los árboles de decisión para identificar valores atípicos/anomalías en datos de series temporales.

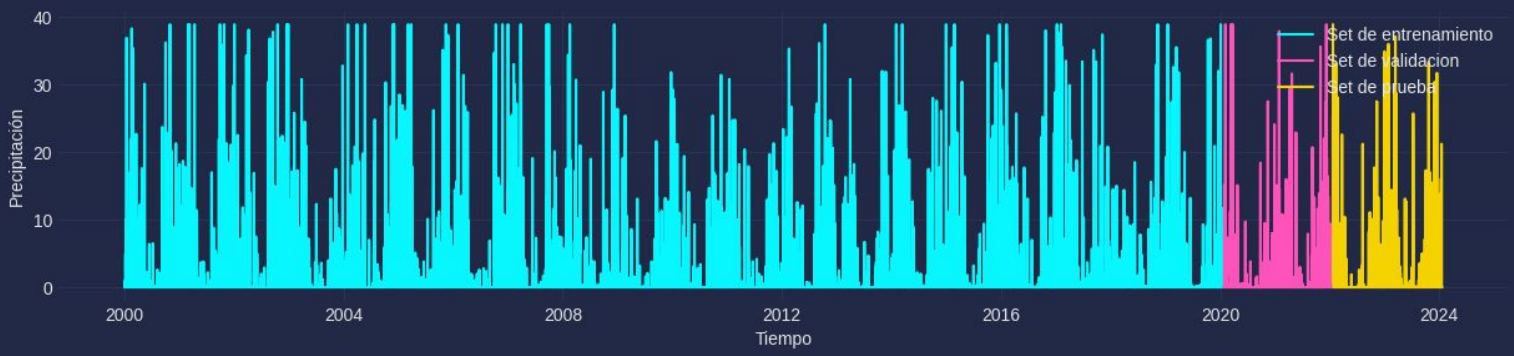
Isolation Forest, como cualquier método de conjunto de árboles, se basa en árboles de decisión. La idea principal, que es diferente de otros métodos populares de detección de valores atípicos, es que Isolation Forest identifica explícitamente anomalías en lugar de perfilar puntos de datos normales.





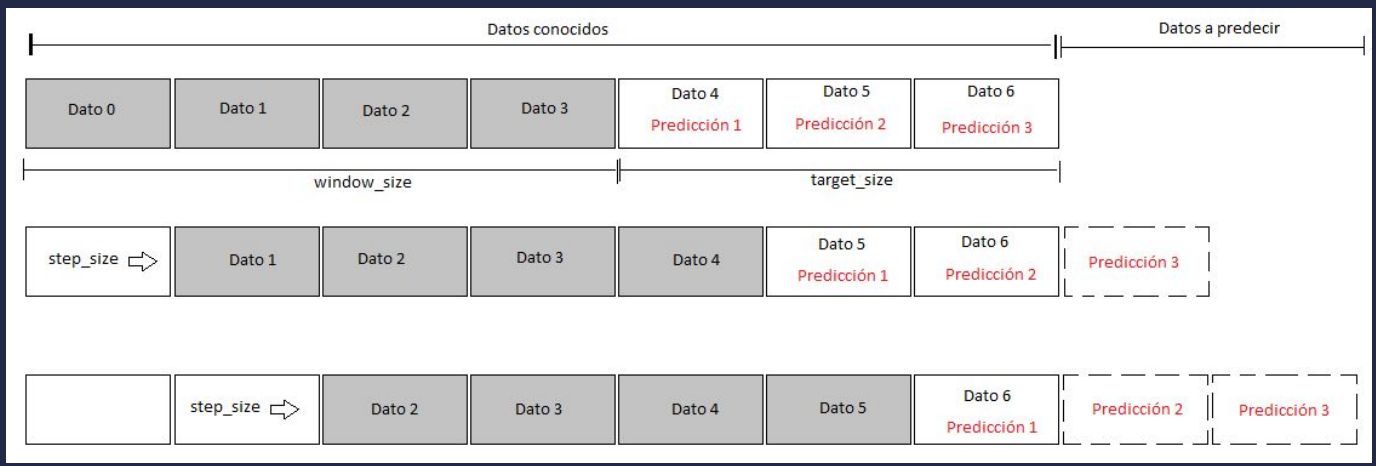
6. Set de entrenamiento, validación y prueba

Tengamos en cuenta que estamos en presencia de un set de datos del tipo ‘Serie Temporal’, por lo que la división del set no puede romper la cronología de eventos, por lo que consideraremos los últimos 4 años como sets de prueba y validación, el resto se utilizará para entrenamiento.



Ventanas temporales

Todos los sets de datos, serán transformados en subsets de largo "window_size” + “target_size". Donde nuestra ventana temporal observará un tiempo de largo "window_size" pasos, de forma tal que cada ventana de largo "window_size" logre predecir "target_size" pasos. La posición temporal de cada ventana estará desplazada "step_size" pasos de su predecesora.

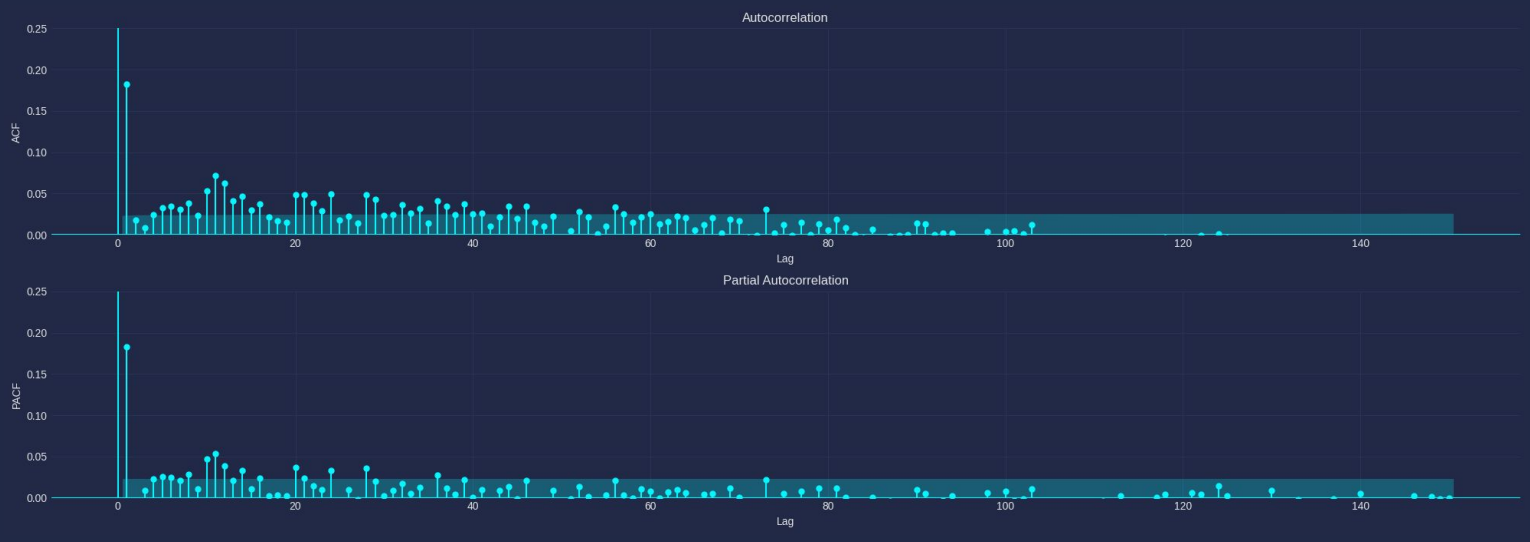


Análisis de autocorrelación

El análisis de autocorrelación ayuda a detectar patrones y comprobar la aleatoriedad en series temporales. El análisis consiste en observar los gráficos de la función de autocorrelación (ACF) y la función de autocorrelación parcial (PACF).

En algunos modelos como el de AutoRegressive-Moving Average (ARMA) este análisis es fundamental para determinar algunos hiperparametros (el orden). Como no abordaremos ninguno de estos modelos, sólo usaremos este análisis para determinar los 'lags' o 'windows_size' que es estadísticamente significante.

Puede verse un área azul en los gráficos ACF y PACF. Esta área azul representa el intervalo de confianza del 95% y es un indicador del umbral de significancia. Eso significa que cualquier cosa dentro del área azul es estadísticamente cercana a cero y cualquier cosa fuera del área azul es estadísticamente distinta de cero.



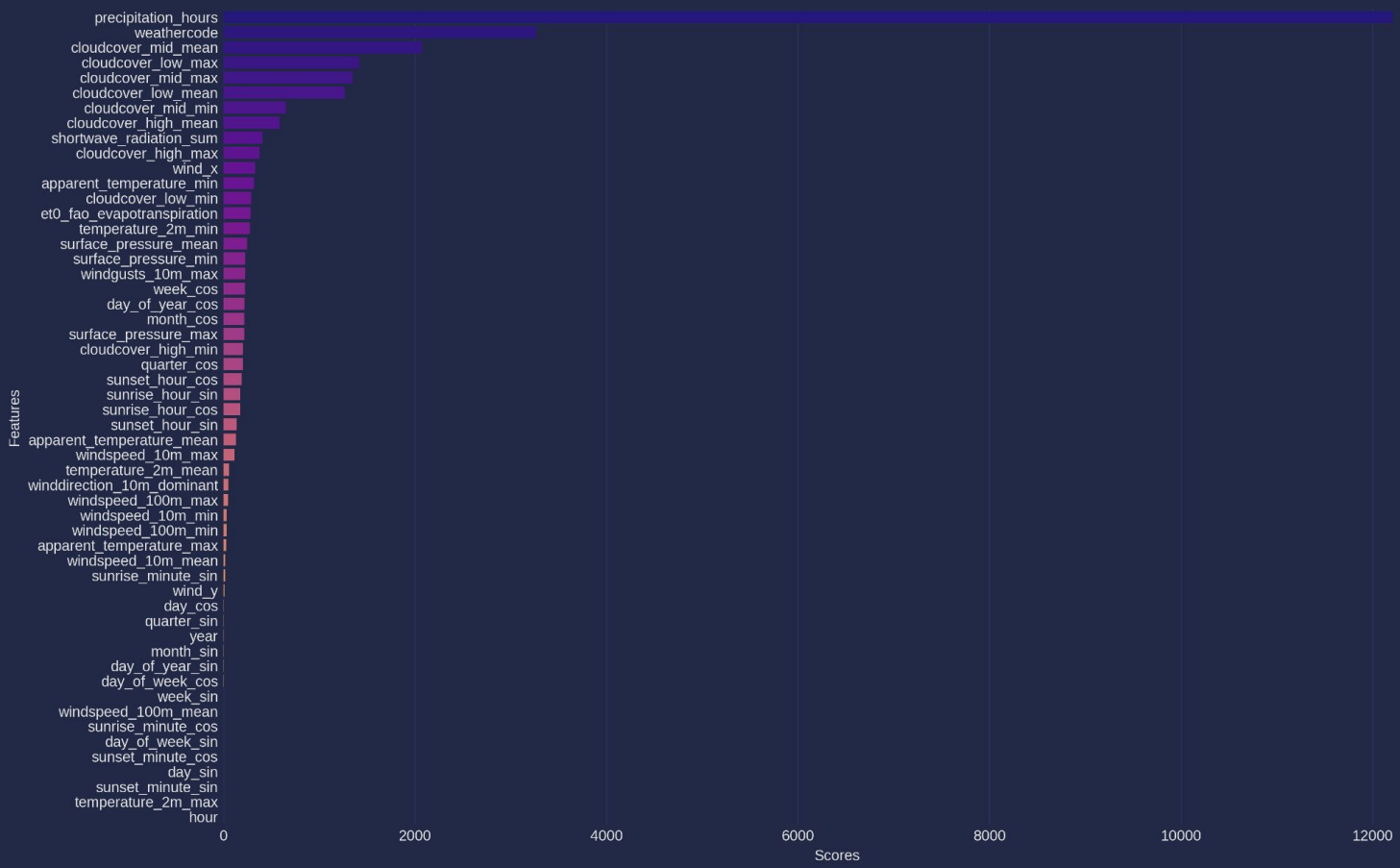
De esta forma determinamos que en el lag 11 hay una fuerte correlación, por lo tanto nuestro 'window_size' será de 11 pasos.

7. Feature Selection

Para la selección de características se utilizó el algoritmo ‘SelectKBest’ configurado con pruebas de regresión lineal univariante y se consideró seleccionar las primeras 40 caracteristicas.

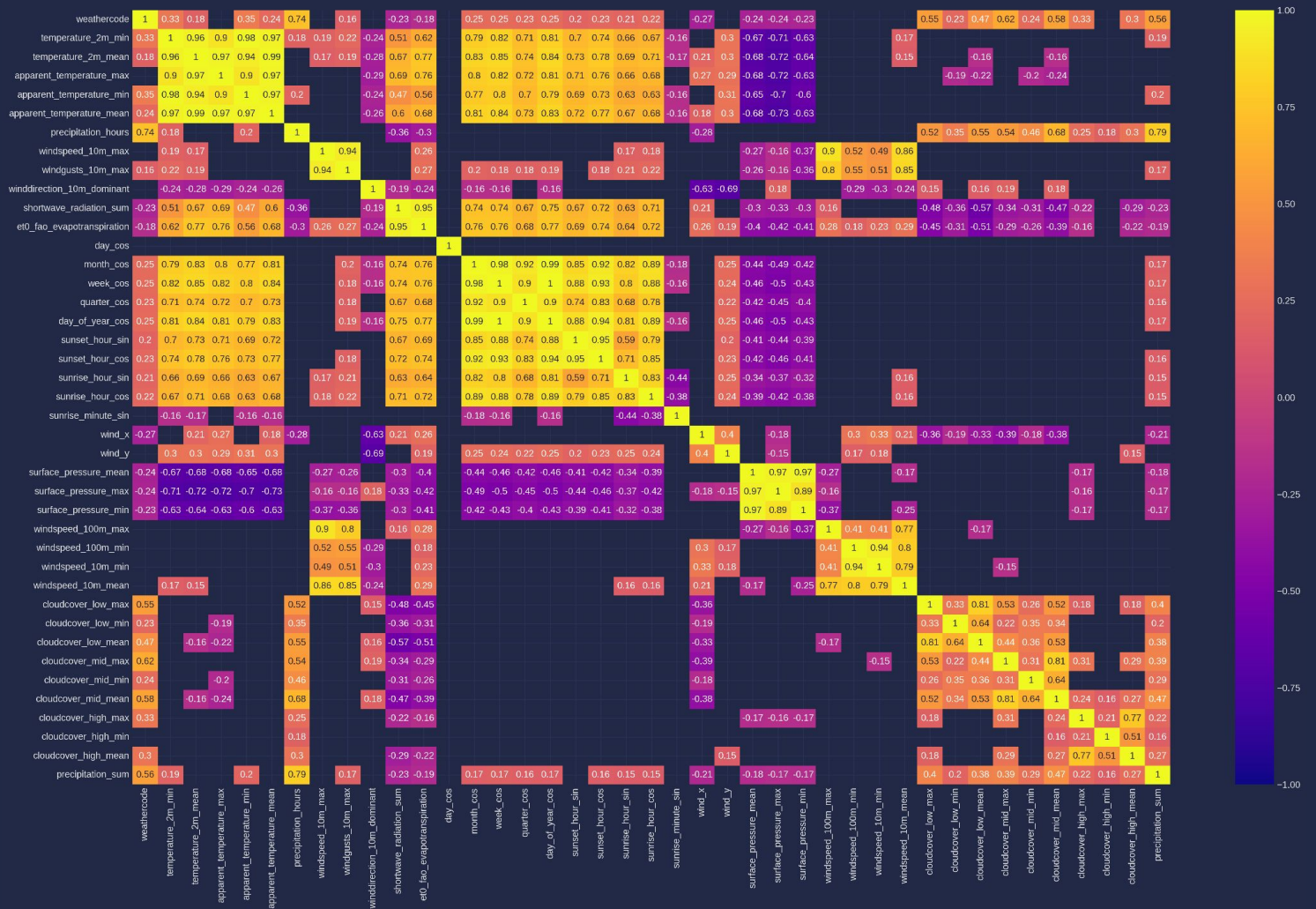
¿Por qué y qué ‘score_func’ usar?

‘f_regression’ es adecuado para problemas de regresión lineal, mientras que ‘mutual_info_regression’ es más adecuado para problemas de regresión no lineal o cuando la relación entre las características y el objetivo no está clara.



Correlaciones y detección de problemas de multicolinealidad

La multicolinealidad ocurre cuando las variables independientes (predictores) en un modelo de regresión están correlacionadas.
Las variables independientes deberían ser eso, independientes. Y esto se debe a que si el grado de correlación entre las variables independientes es alto, no podremos aislar la relación entre cada variable independiente y la variable dependiente (respuesta). Si no podemos aislar los efectos podríamos confundir sus efectos. Para esto se utilizó la métrica denominada 'Variance Inflation Factor'. Con esto reducimos a 28 el total de características.



8. Feature Scaling

El escalado de los datos no siempre implica una mejora en la performance de todos los modelos, pero por lo general suele impactar de forma positiva en las predicciones de los mismos.

La técnica utiliza en este proyecto para el escalamiento de los datos ha sido el 'Min-Max Scaling', esta técnica consiste en una normalización de los valores de un feature entre un rango fijo de 0 y 1.

Se han testeado otro tipos de escalamientos como 'Standard Scaling' y 'Robust Scaling' pero el utilizado en el proyecto fue el que mejor resultados me permitió alcanzar.

9. Modelos

Antes de introducir los modelos utilizados, retomemos los datos de entrenamiento y veamos cual es la cantidad de datos de la que estamos hablando.

```
1  X_train.shape, y_train.shape
```

```
((7310, 11, 28), (7310, 7))
```

Haciendo un cálculo rápido tenemos 7310 * 11 * 28 = 2.251.480 datos que procesar, no es un gran número pero tampoco son pocos datos, es por esto que los modelos que utilizamos deben permitir procesamiento por GPU ya que si no el entrenamiento de los mismos llevaría demasiado tiempo. También debemos tener en cuenta que los modelos que utilicemos deben permite entrada de matrices (11x28) ya que nuestro forecasting será multivariado, también debe permitir salida 'Multi-Output' (7), esto no siempre es posible con todos los algoritmos.

En síntesis, se ha utilizado 1 modelo de aprendizaje automático, uno de los más populares para forecasting de series temporales, el conocido modelo XGBoost que implementa algoritmos bajo el framework Gradient Boosting. También se ha implementado 1 modelo de aprendizaje profundo con TensorFlow mas precisamente un modelo LSTM que es un tipo de red neuronal recurrente (RNN).

Métricas

Nuestro problema es de regresión, así que la métrica de evaluación elegida fue el Mean Absolute Error (MAE) que básicamente mide la diferencia entre dos variables continuas, cuanto más próximo a 0 se encuentre el valor de esta métrica mejor será la performance de nuestro modelo.

Para la comparación entre los modelos, además, se han tenido en cuenta otras métricas tales como el Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), R2 Score y Maximum Residual Error

Entrenamientos

1. XGBoost:

   Este modelo fue entrenado con dos variantes del set de datos, el primer entrenamiento fue llevado a cabo con el set de entrenamiento inicial, se realizaron múltiples optimizaciones al modelo procurando reducir el MAE, con un poco más de 1000 épocas se logró alcanzar el valor óptimo.

   Por otro lado, se consideró entrenar al modelo con un set de datos más reducido, buscando mejorar la eficiencia y los resultados base obtenidos, para esto se utilizó una técnica de 'Feature Reduction' conocida como Análisis de Componentes Principales (PCA) con una explicación de la varianza del 99%, esto me permitió reducir las características totales de 28 a 13. Sin embargo, los resultados no fueron favorables esta reducción si bien mejoró la velocidad del modelo empeoro las predicciones.

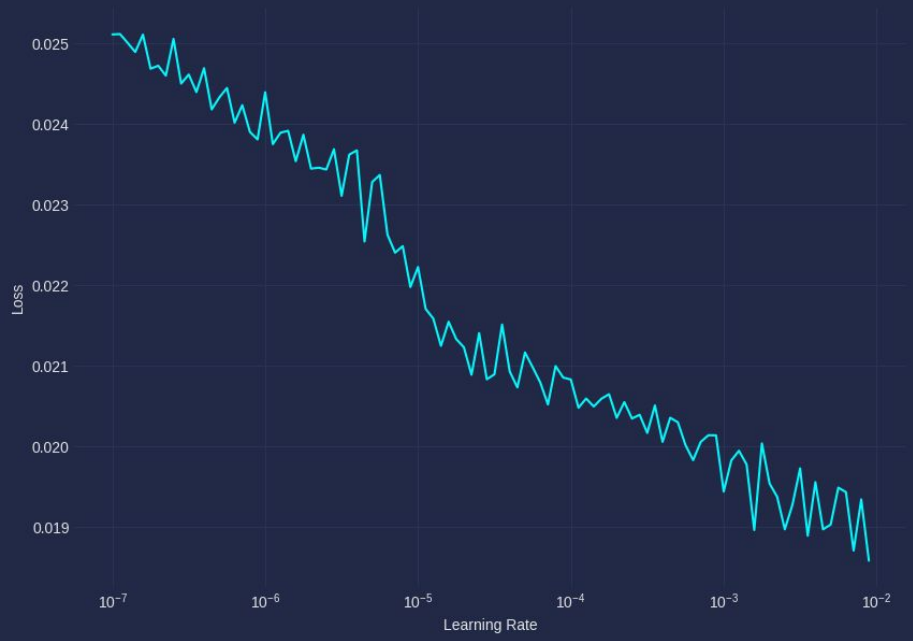
Cabe destacar que este modelo fue entrenado con una estrategia conocida como Early Stopping que es una forma de regularización que se utiliza para evitar el sobreajuste.

Podemos observar que cuando el modelo comenzó a sobreajustar el entrenamiento fue detenido.

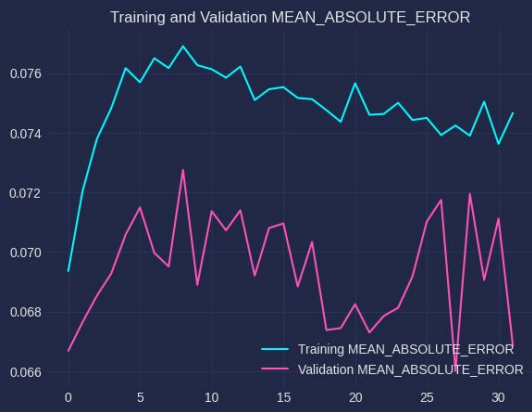
2. LSTM network:

Este modelo a diferencia del anterior, fue entrenado con un único set de datos, el modelo implementado es un Bidireccional Long Short-Term Memory secuencial, este tipo de red neuronal recurrente ha demostrado excelentes resultados en las tareas de forecasting en series temporales multivariables-multioutput.

En toda red neuronal, el Learning Rate (LR) es un hiper parámetro importante a definir, por lo que se ha llevado adelante un scheduler para determinar el LR óptimo.



También al igual que con XGBoost se aplicó la estrategia de Early Stopping para evitar el sobreajuste del modelo.



La iteraciones en el entrenamiento fueron bastantes en búsqueda de optimizar al modelo, se han ajustado tanto los parámetros de las capas ocultas como los hiper parámetros del modelo.
Con alrededor de 30 epocas se consiguieron resultados mejores que los obtenidos por XGBoost.

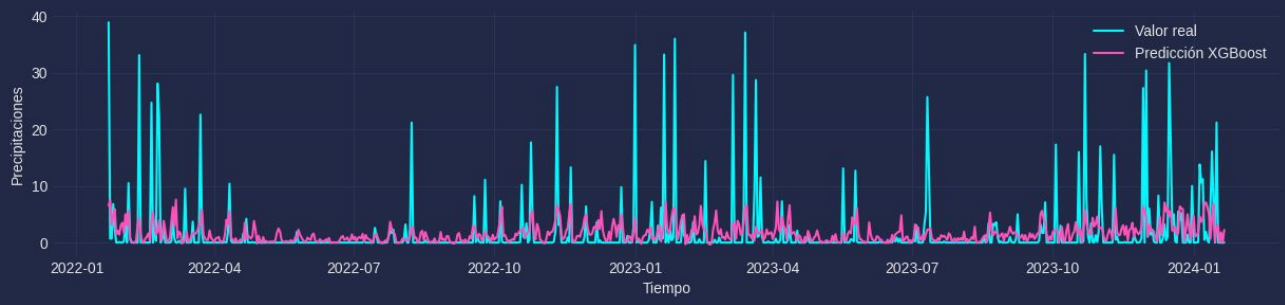
10. Resultados

El resultado más importante que debemos observar son las métricas de las cuales hablamos anteriormente.

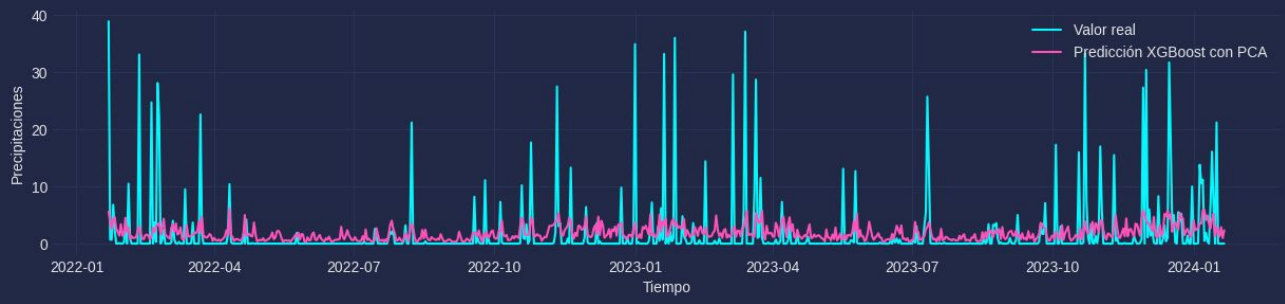
	XGBoost	XGBoost con PCA	LSTM model
MAE	2.314059	2.509714	2.272128
MSE	24.883223	25.765231	27.640180
RMSE	4.988309	5.075946	5.257393
MAPE	3643845815869612.500000	4528253154137266.000000	3238325254708194.500000
R2 Score	0.148936	0.118770	0.054642
Max error	32.373515	33.325772	36.519246

De esta métricas podemos concluir que la mejor performance, guiándonos de la métrica principal de estudio, fue obtenida por la LSTM network.
Visualmente las predicciones lucen de la siguiente forma.

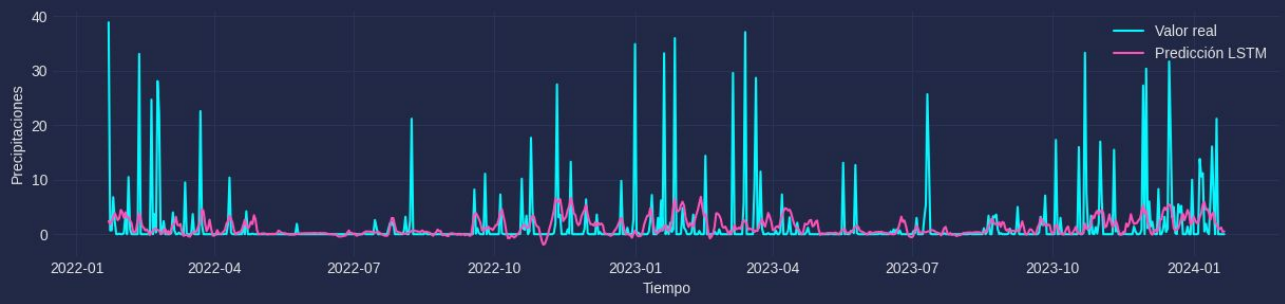
XGBoost



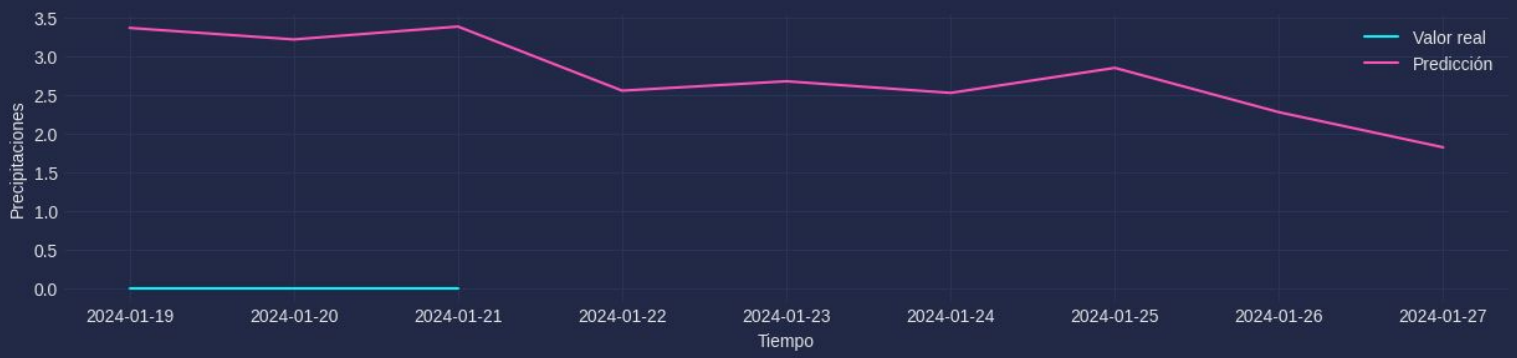
XGBoost con PCA



LSTM



Sin embargo, desde mi punto de vista el modelo que mejor se ajusta a los datos es XGBoost, una de las razones principales es que LSTM a veces predice valores negativos de precipitación, además de que el error residual y el MSE de este modelo es mucho más grande que el de XGBoost. Por eso decidí predecir los siguiente 7 días a partir de la fecha actual (21/01/2024) con este modelo.



11. Modificaciones futuras

Cabe destacar que durante el desarrollo de este proyecto se intentaron muchas técnicas, las cuales tuve que descartar por falta de tiempo, sería interesante poder implementarlas, como la creación de nuevas features basándonos en algoritmos KNN, la implementación de redes neuronales más complejas como los las Conv-LSTM e incluso realizar transfer learning con redes como NBeats dedicas a forecasting de series temporales, se pueden implementar algoritmos como ARMA, AR, y MA.

Alexander Daniel Rios

Visita mis redes (en mi perfil de Github podrás encontrar el notebook de este proyecto)

CODERHOUSE