Análisis de los diferentes modelos predictivos para una serie de tiempo.

Proyecto de Data Science para It Academy

Taida Costa Catalán

ABSTRACT

Este estudio busca analizar varios modelos predictivos precisos y confiables para pronosticar las temperaturas futuras utilizando técnicas de análisis de datos y aprendizaje automático. El objetivo es aprovechar los datos históricos de temperatura para identificar patrones y tendencias, proporcionando información valiosa para la planificación, la optimización de recursos y la toma de decisiones informadas en campos como la agricultura y la gestión de la energía. Se valoraran tres enfoques diferentes evaluando sus ventajas y limitaciones. En resumen, este estudio destaca la importancia de buscar un modelo predictivo eficiente aplicado a datos de temperaturas, con el objetivo de mejorar la toma de decisiones basada en pronósticos confiables y aprovechar al máximo el potencial de los datos disponibles.

1- INTRODUCCIÓN

Una serie de tiempo es una secuencia de datos ordenados en función del tiempo, donde cada observación está asociada con un momento específico. Este tipo de datos se encuentran en muchas áreas, como la economía, las finanzas y la meteorología. El análisis de series de tiempo implica comprender y modelar la estructura temporal de los datos para hacer predicciones sobre su comportamiento futuro.

Problemáticas a la hora de buscar modelos predictivos para series de tiempo:

La predicción precisa de una serie de tiempo presenta varios desafíos. Algunas de las problemáticas comunes incluyen:

1. Tendencias y estacionalidad: Las series de tiempo a menudo exhiben patrones de tendencia y estacionalidad, lo que dificulta la identificación de las relaciones subyacentes y la predicción precisa.
2. Ruido y variabilidad: Las series de tiempo suelen contener ruido y variabilidad aleatoria, lo que puede dificultar la separación de las señales significativas de las fluctuaciones aleatorias.
3. Dependencia temporal: Los valores en una serie de tiempo están correlacionados entre sí debido a la dependencia temporal. Esto debe ser tenido en cuenta al seleccionar y aplicar modelos predictivos.

La razón por la que estas series son importantes es que la mayoría de los modelos de series de tiempo funcionan bajo el supuesto de que la serie es estacionaria. Intuitivamente, podemos suponer que si una serie tiene un comportamiento particular en el tiempo, hay una probabilidad muy alta de que se comportamiento continúe en el futuro. Además, las teorías relacionadas con las series estacionarias son más maduras y más fáciles de implementar en comparación con series no estacionarias. A pesar de que el supuesto de que la serie es estacionaria se utiliza en muchos modelos, casi ninguna de las series de tiempo que encontramos en la práctica son estacionarias. Por tal motivo la estadística tuvo que desarrollar varias técnicas para hacer estacionaria, o lo más cercano posible a estacionaria, a una serie.

2-METODOLOGIA

En este estudio, hemos decidió utilizar el conjunto de datos proporcionados por: " https://opendata-ajuntament.barcelona.cat/data/ca/dataset/temperatures-hist-bcn" se refiere a las temperaturas de la ciudad de Barcelona desde el año 1780 al 2022. Los datos se han descargo en formato CSV y se han procesado con Visual Studio Code en Python3.

Este dataframe contiene datos históricos de temperatura del aire entre los años 1780 hasta 2022 de la ciudad de Barcelona. Contiene 237 filas donde cada fila representa un año y cada columna, en total 13, representa un mes específico, con su respectiva temperatura promedio.

El año esta representado por valores enteros y los meses por valores flotantes. No hay valores nulos.

En el inicio de este proyecto nos hemos fijado estos objetivos:

- Incorporación de las librerías
- Carga del conjunto de datos
- Análisis de los datos
- Modelo Arima: Prueba de Dickey Fuller Aumentada, División de datos para entrenamiento y prueba, Modelo con Auto-Arima, Implementación del modelo, Evaluación del modelo
- Modelo LSTM: Estandarización de los datos, Modelación con LSTM Keras, Evaluación del modelo
- Modelo Prophet: Modelación del modelo Prophet. Evaluación del modelo

3- CONCLUSIONES

Se han utilizado 5 métricas habituales para evaluar y comparar los diferentes métodos de predicción (SARIMA, LSTM Y Prophet), en base a su predicción, capacidad de ajuste y capacidad para explicar la variabilidad de los datos. A continuación, proporciono una breve descripción de cada una de ellas:

- Mean Squared Error (MSE): El Error Cuadrático Medio calcula el promedio de los errores al cuadrado entre las predicciones y los valores reales. Un valor de MSE más bajo indica una mejor precisión del modelo, ya que implica una menor dispersión de los errores.

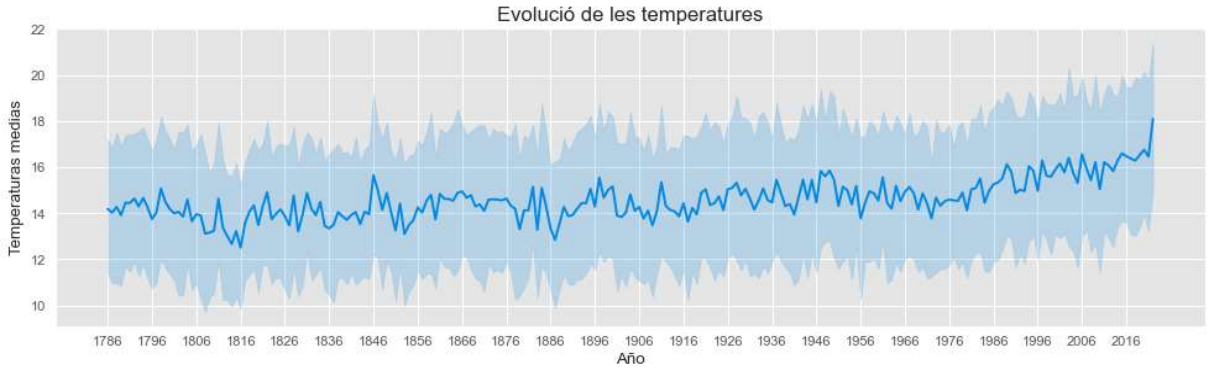
- Mean Absolute Error (MAE): El Error Absoluto Medio calcula el promedio de los errores absolutos entre las predicciones y los valores reales. Es una medida de la magnitud promedio de los errores. Un valor de MAE más bajo indica una mejor precisión del modelo.

- Root Mean Squared Error (RMSE): La Raíz del Error Cuadrático Medio se calcula tomando la raíz cuadrada del MSE. Proporciona una medida del error promedio en la misma escala que los datos originales. Al igual que el MSE, un valor de RMSE más bajo indica una mayor precisión del modelo.

- Mean Absolute Percentage Error (MAPE): El Error Porcentual Absoluto Medio calcula el promedio de los errores porcentuales absolutos entre las predicciones y los valores reales. Mide la magnitud promedio del error como un porcentaje de los valores reales. Un valor de MAPE más bajo indica una mayor precisión del modelo.

- R-squared (R2): El coeficiente de determinación R2 proporciona una medida de qué tan bien se ajustan los valores predichos a los valores reales. R2 varía entre 0 y 1, donde 1 indica un ajuste perfecto del modelo. Un valor de R2 más alto indica una mejor capacidad del modelo para explicar la variabilidad de los datos.

Valoraciones sobre el **modelo ARIMA**. Según la prueba de Dickey Fuller Aumentada, de inicio es una dataframe no estacionario, tal como se puede intuir en el siguiente gráfico, se puede apreciar tiende a ascender en los últimos años.



Se crea una columna diff() con la finalidad de obtener los datos con la variación de datos adyacentes y poder volver a pasar la prueba de Dickey_Fuller, que en esta segunda vez, su valoración es que es estacionaria.

Se implanta el modelo Seasonal ARIMA, es una extensión de ARIMA que admite explícitamente datos de series temporales univariadas con un componente estacional. Agrega tres nuevos hiperparámetros para especificar la autorregresión (AR), diferenciación (I) y media móvil (MA) para el componente estacional de la serie, así como un parámetro adicional para el período de la estacionalidad.

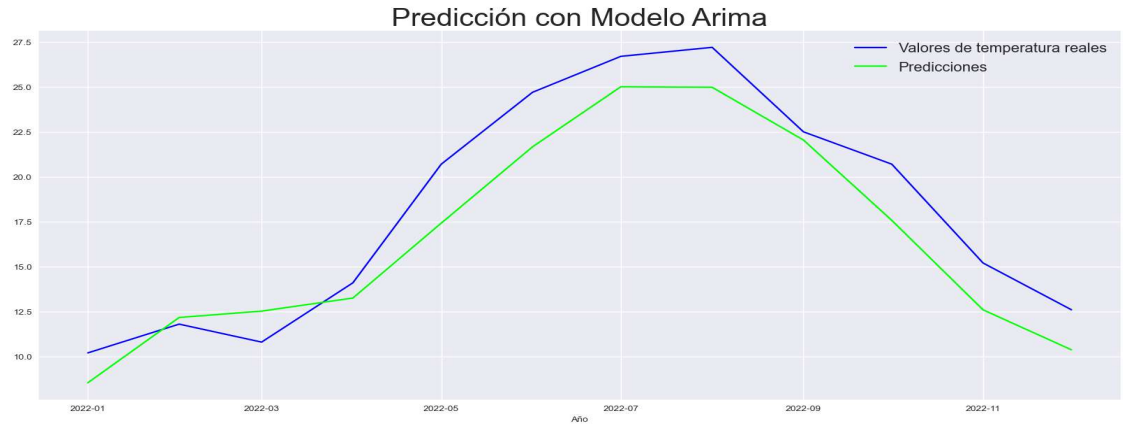
Dividimos el dataframe en train test, con la intención de poder comparar los resultados de los diferentes modelos con valores reales del último año que tenemos constancia.
Fechas datos_train : 1786-01-01 00:00:00 --- 2021-12-01 00:00:00 (n=2832)
Fechas datos_test : 2022-01-01 00:00:00 --- 2022-12-01 00:00:00 (n=12)

Nuestros indicadores de evaluación para el modelo Arima son:

Evaluation metric results: ARIMA
MSE is: 4.674520995909241
MAE is:1.9373197066149803
RMSE is:2.162064059159497
MAPE is:11.331498672440492
R2 is:0.8734506791704397

Basado en estos resultados, el modelo ARIMA parece tener un buen desempeño, con valores relativamente bajos para el MSE, MAE y RMSE, lo que indica una buena precisión en las predicciones. Además, el MAPE es razonablemente bajo, lo que sugiere que el modelo tiene una buena capacidad para predecir los valores correctamente en términos porcentuales. El valor alto del R2 (0.8734) indica que el modelo explica una gran parte de la variabilidad en los datos.

En general, estos resultados sugieren que el modelo ARIMA es eficaz para predecir la serie de tiempo de temperaturas y tiene un buen rendimiento en términos de precisión y capacidad de ajuste. Gráficamente, también se intuye que es un buen modelo de predicción para una serie de tiempo como la nuestra.



El modelo LSTM (Long Short-Term Memory) red neuronal recurrente (RNN) diseñado para capturar y modelar relaciones a largo plazo en secuencias de datos. A diferencia de las RNN tradicionales, que pueden tener dificultades para recordar información de eventos pasados a medida que la secuencia se vuelve más larga, las celdas LSTM están diseñadas específicamente para superar este problema.

Estas puertas permiten que el modelo LSTM retenga información relevante a largo plazo y evite la degradación del gradiente en el entrenamiento.

Se ha utilizado la biblioteca Keras de aprendizaje profundo de alto nivel que proporciona una interfaz fácil de usar para construir y entrenar modelos de redes neuronales, incluyendo modelos LSTM. Keras es ampliamente utilizado debido a su sintaxis intuitiva y su capacidad para ejecutar rápidamente modelos en múltiples plataformas.

Evaluation metric results:LSTM
MSE is : 7.821480482432121
MAE is : 2.5031830822428063
RMSE is : 2.7966909880128195
MAPE is : 14.601050119011013
R2 is : 0.7882557284907605

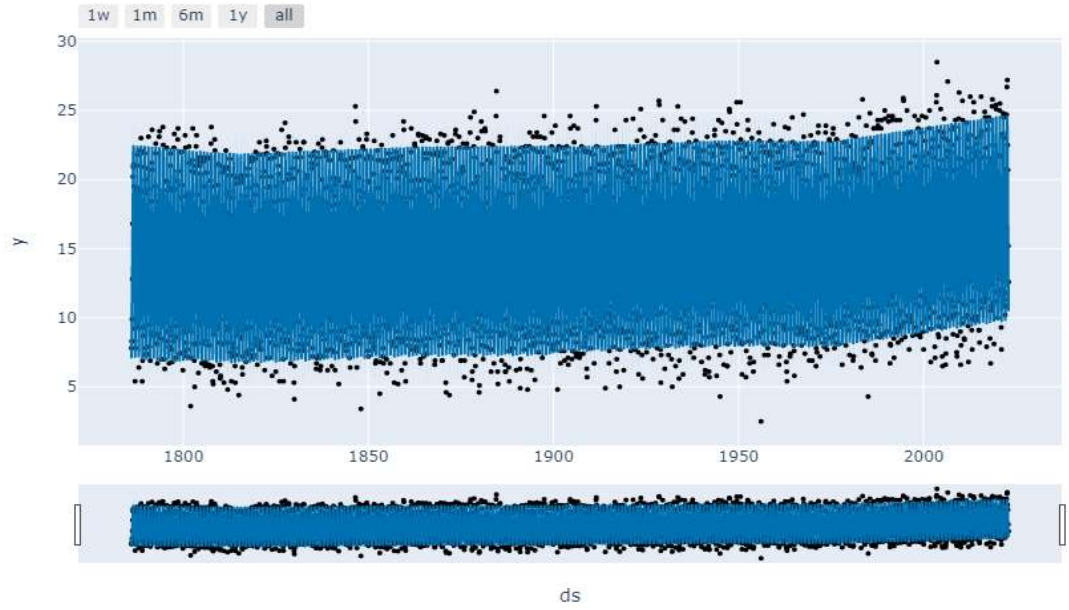
En general, los resultados sugieren que el modelo LSTM tiene un rendimiento razonable. Los valores de MSE, MAE y RMSE son relativamente bajos, lo que indica que las predicciones del modelo están generalmente cercanas a los valores reales. El valor de MAPE de 14.60% sugiere que, en promedio, las predicciones del modelo se desvían aproximadamente un 14.60% de los valores reales. Por último, el valor de R2 de 0.788 indica que aproximadamente el 78.8% de la varianza en la variable objetivo puede ser explicada por el modelo LSTM.

Prophet es una biblioteca de código abierto desarrollada por Facebook para el análisis y la predicción de series de tiempo. Es especialmente conocida por su enfoque intuitivo y fácil de usar, lo que la hace popular entre los usuarios con poca experiencia en el campo de las series de tiempo.

El modelo Prophet se basa en un enfoque aditivo que combina varios componentes para modelar diferentes patrones presentes en las series de tiempo. Estos componentes incluyen tendencia, estacionalidad y efectos de días festivos. Prophet también puede manejar cambios en las tendencias a lo largo del tiempo y manejar valores faltantes en los datos.

Una de las características clave de Prophet es su capacidad para automatizar muchos de los pasos necesarios para el análisis de series de tiempo, como la detección de cambios de tendencia y la selección automática de componentes relevantes. Además, Prophet ofrece una amplia gama de funciones y opciones para personalizar y ajustar el modelo a las necesidades específicas de cada serie de tiempo.

La implementación de Prophet es bastante sencilla. Los pasos típicos incluyen la preparación de los datos de la serie de tiempo, la instanciación del modelo Prophet, el ajuste del modelo a los datos de entrenamiento y la generación de pronósticos utilizando el modelo ajustado. Prophet también proporciona herramientas para visualizar los resultados y evaluar el rendimiento del modelo.



Evaluation metric results:Prophet

MSE is : 4.06043855684281

MAE is : 1.7846236948827119

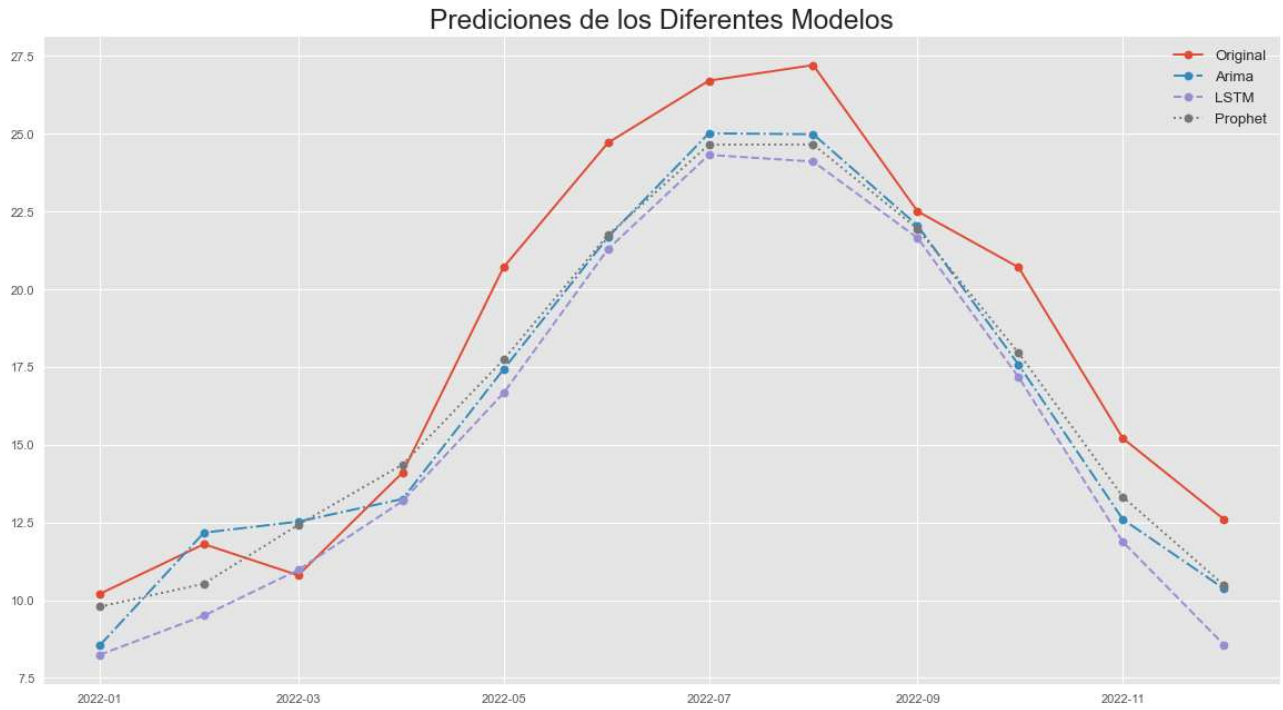
RMSE is : 2.015052991075622

MAPE is : 10.004181981951

R2 is : 0.8900752093982905

En general, los resultados sugieren que el modelo Prophet tiene un buen rendimiento. Los valores de MSE, MAE y RMSE son relativamente bajos, lo que indica que las predicciones del modelo están generalmente cercanas a los valores reales. El valor de MAPE de 10.004% sugiere que, en promedio, las predicciones del modelo se desvían aproximadamente un 10.004% de los valores reales. Por último, el valor de R2 de 0.8900752093982905 indica que aproximadamente el 89.01% de la varianza en la variable objetivo puede ser explicada por el modelo Prophet.

Se puede concluir que los tres modelos evaluados en este proyecto son buenos modelos predictivos para una serie temporal como la presente en este proyecto. El modelo Arima ha obtenido mejores resultados con poco margen respecto el resto de los modelos.



https://www.cienciadedatos.net/documentos/py27-forecasting-series-temporales-python-scikitlearn.html
https://github.com/FrancisArgnR/SeriesTemporalesEnCastellano
https://relopezbriega.github.io/blog/2016/09/26/series-de-tiempo-con-python/
https://www.analyticsvidhya.com/blog/2021/07/introduction-to-time-series-modeling-with-arima/
https://thecleverprogrammer.com/2022/10/17/weather-forecasting-using-python/