

REUNIÓN TFG



Preguntas e ideas

Preguntas:

1. ¿Hago el TFG en inglés?

Lo primero es saber si le parece bien a mi tutor. Lo segundo es saber si merece la pena. Lo tercero es saber si hay que hacer algo distinto en el papeleo.

2. ¿Variables?

- **id**: Identificador único para cada registro.
- **rid**: Identificador para ¿??.
- **cid**: Identificador para ¿??.
- **date**: Fecha y hora de la medida en formato MM/dd/yy hh:mm:ss.
- **MM, dd, yy, hh, mm, ss**: Componentes individuales de la fecha y hora.
- **lid**: Identificador para ¿??.
- **t1**: Temperatura (°C).
- **rh1**: Humedad relativa (%).
- **vpd1**: ¿??
- **dd1**: ¿??
- **lux**: Cantidad de luz en lúmenes.
- **wtlux, alux**: Variables relacionadas con la luz ¿??
- **st1**: Temperatura del suelo (°C).
- **p1**: ¿??.
- **ec1**: Electroconductividad (S/m).
- **vwc1**: ¿??.
- **ecb1, ecp1**: ¿??.
- **st2, p2, ec2, vwc2, ecb2, ecp2**: Variables similares a las anteriores, pero para una segunda ubicación o sensor. ¿??
- **st3, p3, ec3, vwc3, ecb3, ecp3**: Variables similares a las anteriores, pero para una tercera ubicación o sensor. ¿??
- **diam1**: Diámetro (mm).
- **par**: ¿??
- **BV**: ¿??.
- **dli**: ¿??
- **Vardiam, vardiam5 vardiam10, E14 E15** (las que no entiendo de las de Andres)

* En rojo es lo que no entiendo

* En amarillo fosforito está lo que usó Andrés

Sugerencias:

1. Calcular las entradas para maximizar el crecimiento de la planta.

Entiendo que lo que buscamos es estudiar distintas formas para el “Pronóstico de Series de Tiempo”, en nuestro caso predecir el crecimiento de una planta dadas unas entradas. Primero se hizo con Modelos Estadísticos, después con redes neuronales, y después iría yo con otra alternativa.

Por otra parte, a la hora de emplear esto en una industria, supongo que lo que podría ser beneficioso maximizar el crecimiento de la planta para aumentar la producción. Por lo que tendría sentido añadir una forma de controlar el crecimiento variando las entradas.

Aun así, supongo que esta idea no es muy buena, se sale de lo que buscamos aprender con este TFG.

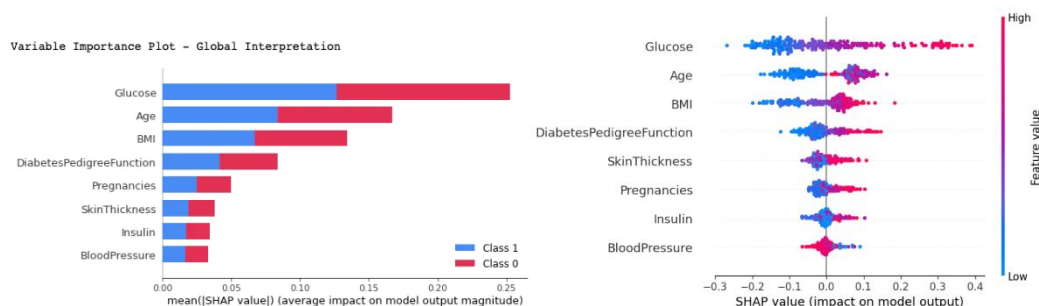
2. Explicabilidad.

La explicabilidad de un modelo de red neuronal es muy útil para comprender cómo toma decisiones y qué características influyen en sus predicciones. Las ventajas de la explicabilidad podrían ser:

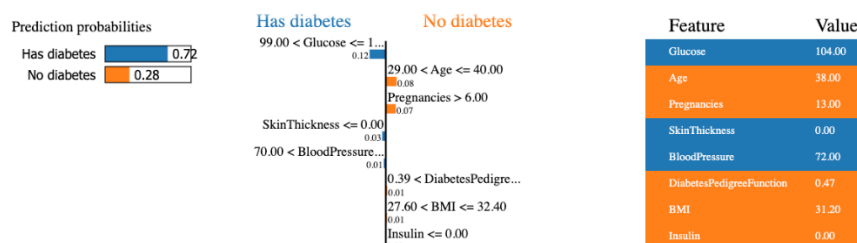
- ¿Cuál de las entradas son más influyentes? Identificar esto puede proporcionar información valiosa.
- Confianza y transparencia: La explicabilidad también aumenta la confianza en el modelo. Si se plantea implementar este modelo en un entorno práctico las explicaciones claras y transparentes ayudan a las personas a comprender y confiar en las predicciones.
- Detectar sesgos y errores: Al analizar la explicabilidad, puedes identificar posibles sesgos o errores. Por ejemplo, si descubres que el modelo se basa demasiado en una sola característica y pasa por alto otras.

Para ello existen varias herramientas como:

- SHAP:



- LIME:



3. Solo se ha usado redes neuronales LSTM, CNN, y

Algunas alternativas que podríamos probar para calcular un modelo son:

- **Máquinas de Vectores de Soporte (SVM):** Pueden ser efectivas para problemas de regresión en series temporales. Podría ser útil su variante diseñada específicamente para series temporales: Máquinas de Soporte Vectorial para Series Temporales (SVR)
- **Redes Neuronales Recurrentes (RNN):** Incluyendo LSTM y GRU, son capaces de capturar dependencias temporales a largo plazo. De estos dos no se ha usado el GRU
- **Transformadores:** Estos modelos, famosos por su uso en procesamiento del lenguaje natural, también se han aplicado con éxito a series temporales. Los Time2Vec y TSTransformer son ejemplos de implementaciones de transformers para datos secuenciales

La idea que más me gusta son los Transformers.

Common time series models:

Old school:

- **ARMA:** Autoregressive Moving Average
- **ARIMA:** Autoregressive Integrated Moving Average
- **SARIMA:** Seasonal Autoregressive Integrated Moving Average
- **SARIMAX:** Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors
- **VARMA:** Vector Autoregression Moving Average
- **VARMAX:** Vector Autoregression Moving Average with Exogenous Regressors
- **SES:** Simple Exponential Smoothing
- **HWES:** Holt-Winter's Exponential Smoothing
- **ARCH:** Autoregressive Conditional Heteroskedasticity
- **GARCH:** Generalized Autoregressive Conditional Heteroskedasticity

Modern methods:

- LSTM/RNN
- Resnet
- ROCKET
- FCN
- Matrix Factorization w/DTW
- XGBoost
- Prophet
- TS-CHIEF
- HIVE-COTE

Resnet: ResNet es una red neuronal profunda que se caracteriza por su capacidad para entrenar modelos con muchas capas (también conocidas como profundidad). Su nombre proviene de la idea central: aprender funciones residuales en relación con las entradas de las capas. A diferencia de las redes neuronales convencionales, que pueden sufrir degradación en la precisión a medida que se agregan más capas, ResNet permite construir redes más profundas sin este problema

ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels

...

4. Ajustar hiperparámetros

En el TFG de Andrés se ha experimentado con distintos hiperparámetros y se ha observado cómo influían en los resultados. Pero no se ha estudiado cuáles serían los hiperparámetros óptimos.

Existen varios algoritmos de ajuste de hiperparámetros, aunque los tipos más utilizados son la optimización bayesiana, la búsqueda por cuadrícula y la búsqueda aleatoria.

- **Optimización bayesiana**

La optimización bayesiana es una técnica basada en el teorema de Bayes, que describe la probabilidad de que ocurra un evento de acuerdo con el conocimiento actual. Cuando esto se aplica a la optimización de hiperparámetros, el algoritmo crea un modelo probabilístico a partir de un conjunto de hiperparámetros que optimiza una métrica específica. Además, utiliza análisis de regresión para elegir de forma iterativa el mejor conjunto de hiperparámetros.

- **Búsqueda por cuadrícula**

A través de la búsqueda de cuadrícula, se especifica una lista de hiperparámetros y una métrica de rendimiento, y el algoritmo trabaja con todas las combinaciones posibles para determinar la opción más adecuada. La búsqueda por cuadrícula funciona bien, pero es relativamente tediosa e intensiva a nivel computacional, sobre todo con números elevados de hiperparámetros.

- **Búsqueda aleatorizada**

Aunque se basa en principios similares a los de la búsqueda por cuadrícula, la búsqueda aleatorizada busca grupos seleccionados de hiperparámetros de manera aleatoria en cada iteración. Funciona bien cuando un número relativamente pequeño de hiperparámetros determinan de manera principal el resultado del modelo.

5. Sostenibilidad

Esto es una tontería un poco hippy.