Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey


TC3006C: Inteligencia artificial avanzada para la ciencia de datos I

Alfredo Esquivel Jaramillo
Antonio Carlos Bento
Frumencio Olivas Alvarez
Hugo Terashima Marín
Jesús Adrián Rodríguez Rocha
Julio Antonio Juárez Jiménez
Mauricio González Soto

# Reporte sobre el desempeño de modelo predictor de edad

Francisco Salas Porras

A01177893

07 / 09 / 2024

# Introducción

Para éste portafolio se trabajó en el entrenamiento de un modelo de tipo random forest regression el cual pudiera predecir la edad de un paciente dado su historial de salud y estilo de vida. El principal propósito es demostrar las habilidades y experiencia en *machine learning* obtenida tras finalizar el primer periodo de *Inteligencia artificial avanzada para la ciencia de datos*. Se utilizarán las librerías de *matplotlib* para graficación, *sklearn* para todo lo de *machine learning*, *pandas* para guardado de los datasets y *numpy* para definición de arreglos y procesos matemáticos.

# Limpieza del dataset

El dataset utilizado para el entrenamiento y prueba del modelo fue proporcionado por los usuarios *M Abdulah* y *Shahzaib Yaqoob*, el cual proporciona 25 diferentes tipos de datos sobre salud y estilo de vida de 3000 individuos. [1]

Éste contiene:

- Género
- Altura
- Peso
- Presión arterial
- Índice de masa corporal
- Nivel de glucosa
- Densidad ósea
- Agudeza visual
- Habilidad de escucha
- Nivel de actividad física
- Frecuencia de fumar
- Consumo de alcohol
- Dieta
- Enfermedades crónicas
- Uso de medicamento
- Historial familiar
- Funciones cognitivas
- Estatus de salud mental
- Patrones del sueño
- Niveles de estrés
- Exposición a contaminación
- Exposición al sol
- Nivel de educación
- Nivel de ingreso
- Edad

Debido a que el propósito del portafolio va más enfocado al área de *machine learning*, solo se mencionará brevemente que en esta parte se limpiaron y transformaron los datos para poder ser procesados por el modelo.

## Entrenamiento y pruebas

El primer paso para poder realizar cualquier entrenamiento de modelo es dividir el dataset en 4 grupos. Primero se realiza la división entre las variables independientes (también referidas como *features* ó la matriz X) y la variable dependiente (también referida como *target* ó el vector y). Las variables independientes van a ser las que serán analizadas y utilizadas por el modelo para que encuentre cómo llegar a la variable dependiente resultante. Teniendo separadas las 2 variables, ahora se realiza una división de datos de entrenamiento y datos de prueba en cada una. Los datos de entrenamiento serán utilizados para proporcionar al modelo y que éste pueda aprender sobre cuáles *features* dan cierto *target*. Por otro lado, los datos de prueba serán reservados para demostrar la efectividad de nuestro modelo con datos que no ha visto antes.

Una vez se tiene divididos los datos, se toman las variables independientes y se escalan para poder tener un mejor rendimiento y reducir el rango en el cual se extienden los datos. Finalmente con los datos escalados se puede hacer un entrenamiento inicial del modelo. Como modelo inicial se definió que contenga los parámetros:

- n_estimators=100
- criterion='squared_error'
- min_samples_split=2
- random_state=3

Estos hiper-parámetros indican que el *random forest* tendrá 100 árboles, aprenderá con una función de costo de la media del error cuadrado, se dividirá solo si tienes mínimo 2 *samples* y está predefinido con un estado random de 3 para asegurar reproducibilidad.

El modelo es luego ajustado a los datos de entrenamiento para poder cambiar los parámetros de los árboles y poder predecir los datos proporcionados. Con los parámetros ajustados, se realiza una predicción de los valores de entrenamiento y prueba para poder observar el sesgo y varianza. El entrenamiento tuvo una media del error cuadrado de 4.14 y un Accuracy de 99.96%. La prueba tuvo una media del error cuadrado de 29.18 y un Accuracy de 99.9%. Esto da a entender que hay una menor cantidad de sesgo (*bias*) y una mayor cantidad de varianza (*variance*). Estos errores indican un overfitting del modelo.

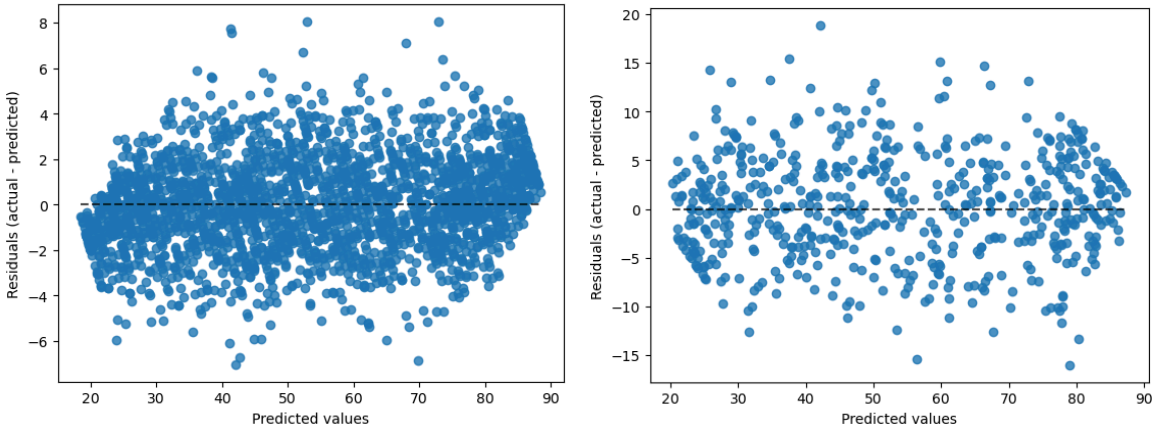


Figura 1. Gráfica de error con entrenamiento no-validado en datos de entrenamiento (izq.) y en datos de prueba (der.)

## Validación y pruebas

Debido al error en varianza previamente mencionado, se intentará reducir la complejidad del modelo. Como primer acercamiento se optó por utilizar la técnica de random search cross validation, la cual utiliza un rango de parámetros especificados para poder ser seleccionados aleatoriamente y ser probados en diferentes dobleces del dataset de entrenamiento. Los parámetros optimizados luego son utilizados para ajustar un nuevo modelo a los datos y poder predecir la variable dependiente de nuevo. El entrenamiento tuvo una media del error cuadrado de 1.05 y un Accuracy de 99.98%. La prueba tuvo una media del error cuadrado de 29.02 y un Accuracy de 99.9%. Aunque hubo una ligera mejora, no es lo suficientemente notoria.

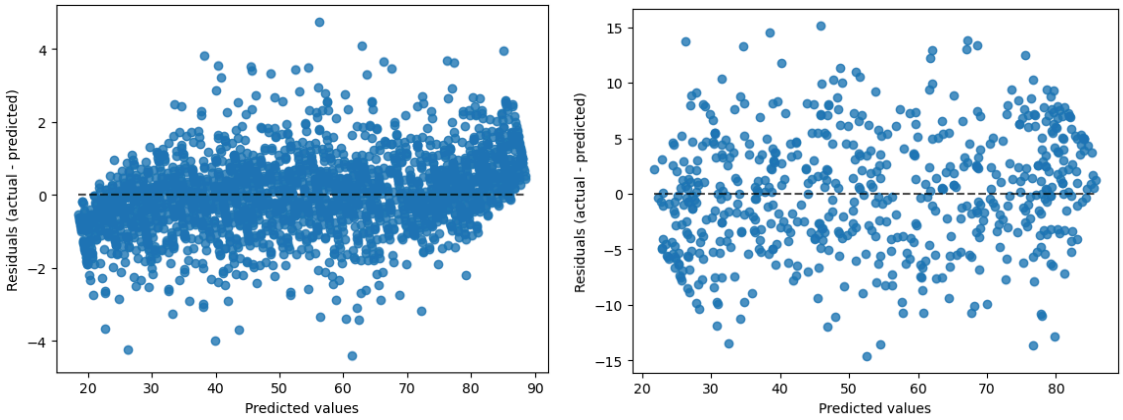


Figura 2. Gráfica de error con entrenamiento validado con random search en datos de entrenamiento (izq.) y en datos de prueba (der.)

Como segunda opción de validación se buscó aplicar un simple grid search cross validation. A diferencia del random search, el grid search va por cada una de las combinaciones de parámetros posibles, además de ser aplicada a cada uno de los dobleces de *cross validation*. Los hiper-parámetros utilizados aquí fueron seleccionados en consideración de los parámetros obtenidos por el anterior modelo. El entrenamiento tuvo una media del error cuadrado de 2.11 y un Accuracy de 99.97%. La prueba tuvo una media del error cuadrado de 34.86 y un Accuracy de 99.89%.

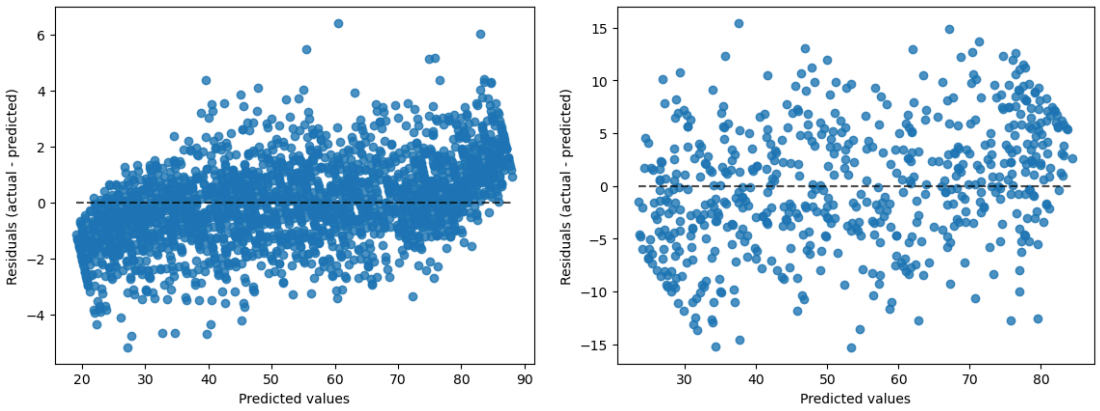

Figura 3. Gráfica de error con entrenamiento validado con grid search en datos de entrenamiento (izq.) y en datos de prueba (der.)

## Conclusiones

Aunque se redujo levemente el error al realizar la validación, no pude reducirlo mucho más de lo que inicialmente obtuve. Hubo algunas consideraciones que posiblemente pudieran haber ayudado al modelo durante la limpieza de datos, pero debido al tiempo y enfoque del reporte no se pudo implementar. Sin embargo, personalmente pude comprender mucho mejor el funcionamiento de la librería de scikit-learn y las diferentes herramientas que aporta. Aplicando también la teoría aprendida durante clase.

## Referencias

1. abdullah, M. (2024). Human Age Prediction Synthetic Dataset. Retrieved September 8, 2024, from Kaggle.com website: https://www.kaggle.com/datasets/abdullah0a/human-age-prediction-synthetic-dataset