IA para la predicción de la edad de los Abalones

Fabián González Vera A01367585

I. Introducción

El abulón o abalón es un molusco gasterópodo de la familia Haliotidae. El cálculo de la edad de uno estos moluscos se lleva a cabo a través de un proceso laborioso y tedioso el cual consiste en cortar a un ejemplar por la espira y se lija hasta exponer los anillos internos, posteriormente estos se cuentan con la ayuda de un microscopio. Se puede usar otras medidas más fáciles de obtener para predecir la edad aproximada. Esta información puede ser usada con el fin de apoyar la conservación de las especies al poder obtener estadísticas como el cambio del promedio de edad, adaptaciones a cambios en el ambiente, etc.[1]

Con el fin de realizar estas predicciones se realiza la implementación del algoritmo de Random Forest de Regresión

El algoritmo de Random Forest es un modelo conformado por la combinación de múltiples Arboles de decisión individuales, en estos se utiliza la técnica de Bagging la cual consiste en generar muestras aleatorias de los datos y usarlas para entrenar modelos independientes, de los cuales se usa el promedio del resultado de estos para producir una predicción más precisa. Este algoritmo puede ser utilizado tanto para problemas de regresión como para problemas de clasificación.

Este algoritmo es popular debido a que reduce el riesgo de Overfitting esto debido a la toma de muestras de los datos; ofrece flexibilidad debido a que puede usarse para problemas de regresión y clasificación; y es fácil de determinar la importancia de las características que tienen mayor impacto en las predicciones.[2]

II. Dataset

El dataset proviene del repositorio UCI Machine Learning de la universidad de California, Irvine[3], el cual mantiene alrededor de 647 datasets para Machine Learning. El dataset usado para este proyecto contiene las medidas de ciertas características físicas de 4177 ejemplares de abulones, dentro de estas se encuentra el sexo, longitud, diámetro, altura, pesos bajo diferentes condiciones y los anillos, estos últimos son usados para medir la edad.

Para el propósito de este proyecto se usaron las variables de:

- Sex_I
- length
- Diameter
- Height
- Whole_weight
- Shell_weight

Y el dataset se dividió en 70% para train y 30% para test.

III. Modelo

El modelo de Random Forest se implementó utilizando el Framework de **scikit-learn**[4], utilizando la función de **GridSearchCV** para realizar una búsqueda exhaustiva sobre un conjunto de parámetros para obtener los hiperparametros óptimos para el modelo. Dentro de esta se configuro para integrar el proceso de Cross validation a través del proceso de **Repeated k-fold cross validation** para reducir la varianza de la precisión del modelo.

GridSearchCV funciona a través de generar un modelo por cada combinación posible de los parámetros que recibe a partir de un diccionario y los evalúa usando un método de Cross-Validation que en este caso es Repeated k-fold cross validation y almacena la configuración del modelo que incluye la mejor combinación de parámetros. Su principal limitante es que solo realiza la búsqueda dentro del conjunto de parámetros que fueron ingresados.[5]

Repeated k-fold cross validation es una variación del proceso de k-fold cross validation en el cual el proceso se repite *n veces* y donde se vuelve a mezclar la muestra resultando en un Split diferente de la muestra. El proceso de k-fold cross validation es un proceso de *resampling* usado para evaluar modelos de machine learning con un numero limitado de datos, consiste en mezclar el dataset, dividirlo en *k* grupos y por cada grupo único se entrena un modelo con los datos restante y se evalúa con el grupo seleccionado, se guarda la puntuación y se descarta el modelo.[6]

Esta implementación dio como resultado que los mejores parámetros son:

- Máxima profundidad: 6
- Numero de características: None
- Máximo número de nodos hoja: 9
- Numero de árboles: 150

IV. Resultados

El promedio de puntaje de la validación cruzada del modelo con los mejores parámetros fue -2.38158

Una Distancia media cuadrática mínima (RSME) de 2.37098

Un coeficiente de determinación R^2 con el dataset de entrenamiento 0.49144

Un coeficiente de determinación R^2 con el dataset de prueba de 0.44555

Esto significa que el modelo explica entre el 49% y el 44% de la variabilidad en los datos.

A través de la librería *mlxtend*[7] se estimó el bias o sesgo, la varianza y el error del modelo utilizando la función *bias_variance_decomp*. La cual calcula el bias como $Bias = y - E[\hat{y}]$, la varianza como $Var = E[(E[\hat{y}] - \hat{y})^2]$ y el error cuadrático medio como $E[MSE] = [Bias]^2 + Var$.[8]

Este cálculo dio como resultado:

Un bias de 18791.084, una varianza de 0.177 y un error de 15.162; dados estos resultados podemos concluir lo siguiente:

- La modelo esta sesgado hacia la información del dataset, esto significa que es un modelo muy rígido

- Tiene una varianza muy baja por lo cual habría cambios pequeños en el modelo si cambiara el dataset

Referencias:

[1] Gluyas-Millán, María Georgina, & Talavera-Maya, Jesús. (2003). Composición por tallas y edades de las poblaciones de abulón Haliotis fulgens y H. corrugata de la zona de Bahía Tortugas, Baja California Sur, México. Ciencias marinas, 29(1), 89-101. Recuperado en 11 de septiembre de 2023, de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-38802003000100009&lng=es&tlng=es.

[2] What is Random Forest? | IBM. (s. f.). https://www.ibm.com/topics/random-forest

[3] Nash,Warwick, Sellers,Tracy, Talbot,Simon, Cawthorn,Andrew, and Ford,Wes. (1995). Abalone. UCI Machine Learning Repository. https://doi.org/10.24432/C55C7W.

[4] API design for machine learning software: experiences from the Scikit-learn Project, Buitinck et al., 2013.

[5] Okamura, S. (2021, 25 diciembre). GridSearchCV for Beginners - towards Data Science. Medium. https://towardsdatascience.com/gridsearchcv-for-beginners-db48a90114ee

[6] Brownlee, J. (2020). A gentle introduction to K-Fold Cross-Validation. MachineLearningMastery.com. https://machinelearningmastery.com/k-fold-cross-validation/

[7] Raschka, S. (s. f.). Mlxtend. http://rasbt.github.io/mlxtend/

[8] Raschka, S. (s. f.). Bias_Variance_decomp: Bias-variance decomposition for Classification and Regression losses - MLXTEND. https://rasbt.github.io/mlxtend/user_guide/evaluate/bias_variance_decomp/