Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

Inteligencia Artificial Avanzada para la Ciencia de Datos I

Módulo 2 - Análisis y Reporte del Desempeño del Modelo

TC3006C

Grupo: 101

Profesor

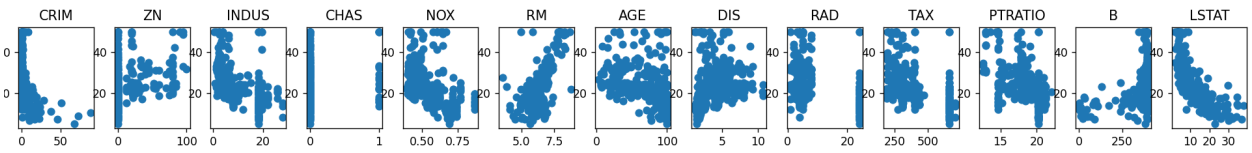
Jorge Adolfo Ramirez Uresti

Presenta

David Damián Galán - A01752785

Fecha de entrega: 11 de septiembre de 2023

El presente documento tiene como objetivo dar una descripción detallada de cómo es que se evalúa el modelo creado anteriormente en la evidencia de implementación de un algoritmo de Machine Learning. En este caso, utilizaremos el algoritmo de la actividad de uso de un Framework, que corresponde al modelo de Decision Tree Regressor en dataset “Boston Housing”. Se eligió este dataset porque contiene datos sobre el precio de las casas para cada registro, además de variables que podrían utilizarse para predecir dicho precio. Por tal razón, utilizamos el modelo de regresión de Decision Tree incluido en la librería de scikit-learn.

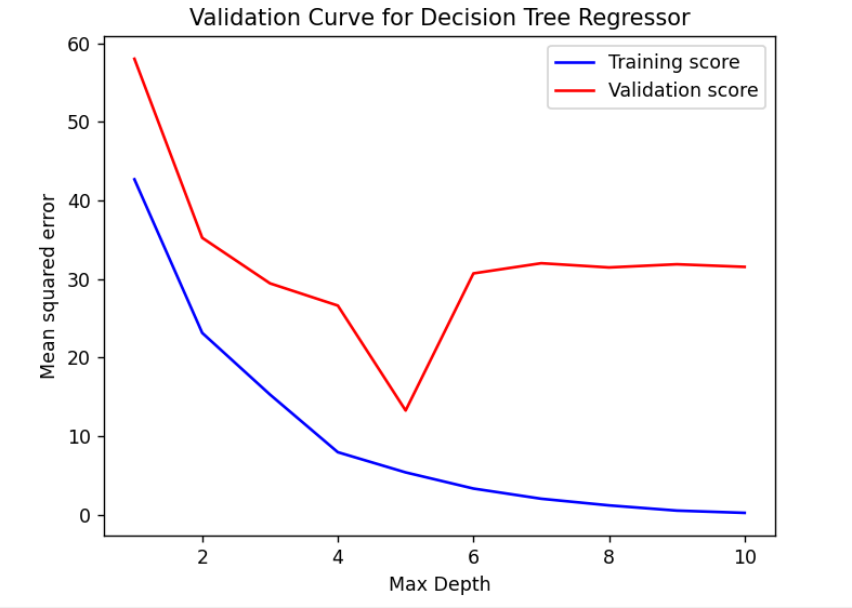


[1] Visualización de las variables contra la variable que queremos predecir (MEDV, valor medio del inmueble). En el eje X se muestra la variable del título para cada gráfica y en eje Y, el valor de MEDV.

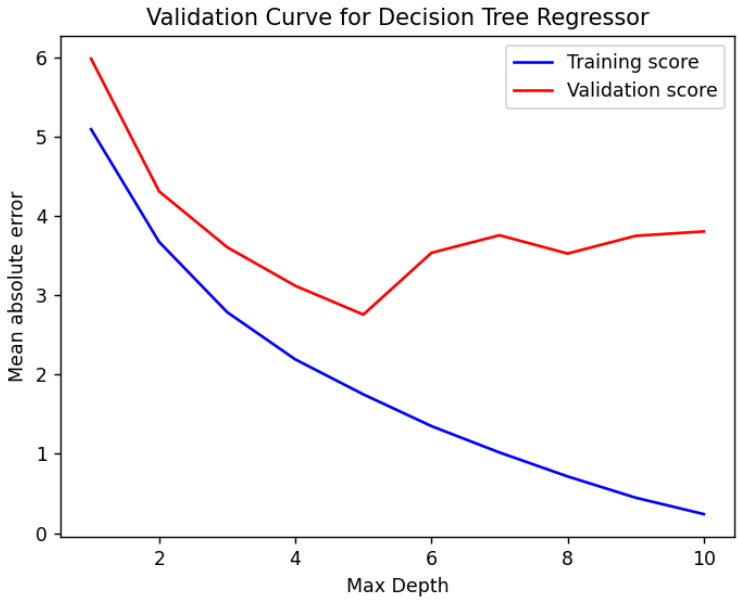
Hiperparámetro que se ajusta

En el algoritmo seleccionado, vamos a ajustar el hiperparámetro de Max Depth, que representa el número máximo de niveles que tiene el árbol.

En nuestra implementación, utilizamos todas las variables como features para tratar de predecir el valor. Para comprobar la efectividad del modelo utilizamos 3 gráficas, que representan el Error Medio Cuadrático (MSE), Error Medio Absoluto (MAE) y Coeficiente de Determinación (R2). En el eje X, se muestra el valor del hiperparámetro y en el eje Y, el valor de cada una de las métricas. Obtenemos los siguientes resultados en una corrida.



[2] Resultados del score MSE.



[3] Resultados del score MAE.

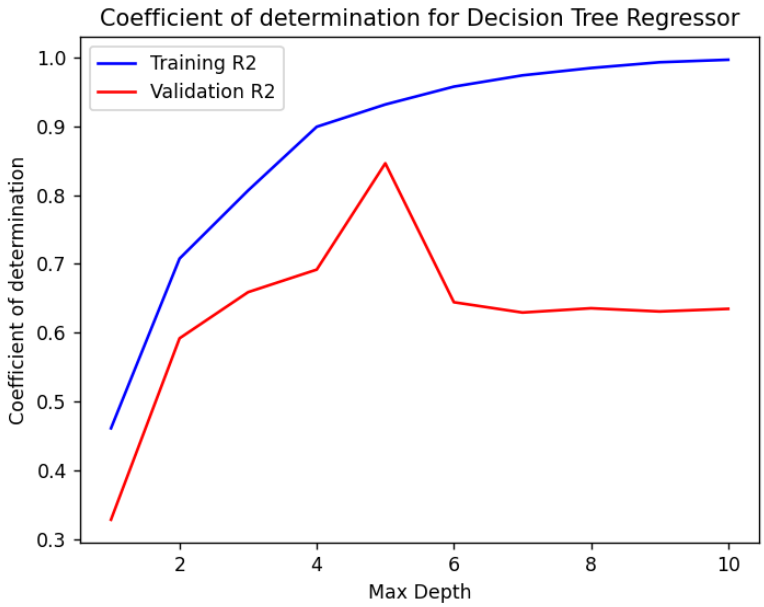
Para las dos primeras gráficas, del Error Medio Cuadrático y del Error Medio Absoluto, observamos que el error de training score es menor que el valor del test score, lo cual es esperado ya que el modelo se entrenó con el primer dataset. Analizando el progreso de ambos valores de error, observamos que entre mayor es el valor de max depth, menor es el valor del error en el

dataset de entrenamiento, pero no ocurre lo mismo en el dataset de validación. En este caso, el error del dataset de pruebas va disminuyendo hasta que llega a un punto en el cual incrementa súbitamente y posteriormente se mantiene.

Podemos explicar este comportamiento de la siguiente manera:

Con valores muy pequeños de max depth, por ejemplo, max depth = 1, el MSE y el MAE tienen valores altos tanto en el dataset de entrenamiento como en el de pruebas. Esto significa que el grado de bias es alto y el de varianza es bajo, donde el término "bias" se refiere justamente al nivel de error entre el valor real y el valor predecido, y el término varianza se refiere la a variabilidad de la predicción cuando se sustituye la muestra de datos por una diferente. En este caso, tenemos un fenómeno de underfitting, lo cual significa que el modelo no ha aprendido suficiente acerca de los patrones y relaciones detrás de los datos. Como tenemos un modelo con underfitting, no se predicen ni los valores del dataset de entrenamiento ni los del dataset de validación.

Ahora, por el lado contrario, con los valores muy altos de max depth, como max depth = 10, nos encontramos con el fenómeno de que ambos errores (MSE y MAE) son muy pequeños para el dataset de entrenamiento pero considerablemente más grandes para el dataset de validación. Esto significa que el modelo sufre de overfitting, lo que quiere decir que el grado de bias es bajo (error muy pequeño en el dataset de entrenamiento), pero la varianza es alta (al introducir el dataset de validación, efectivamente cambiando los datos a los que está expuesto el modelo, las predicciones son de mala calidad). El overfitting también puede interpretarse como que nuestro modelo ha aprendido no solamente de la propia distribución de los datos, sino también del ruido inherente del dataset de entrenamiento, lo que una vez más, afecta las predicciones cuando se introducen nuevos datos.



[4] Resultados del score R2.

Comprobamos nuestros supuestos con una tercera métrica, que es el R2 score. En este caso, un mayor valor indica un mejor ajuste. Al igual que las otras dos métricas, en esta se observa un crecimiento continuo hasta el punto en el que solamente crece en el dataset de entrenamiento y cae en el de validación. Las tres gráficas coinciden en que el mejor valor para el hiperparámetro, en el cual el modelo no sufre de underfitting ni overfitting, es el valor de 5. Esto también es el equivalente a utilizar una técnica de regularización, que tienen como objetivo precisamente que el modelo no sufra de overfitting, "limitando" la capacidad de aprendizaje del algoritmo. Para nuestro caso del parámetro de max depth, básicamente estamos limitando la altura máxima que árbol puede extenderse, lo cual a su vez limita que el árbol se siga extendiendo y se cree un modelo demasiado ajustado al dataset de entrenamiento.

```
FINAL MODEL
*** Max depth value: 5 ***
Mean squared error in test set = 21.08742804330065
Coefficient of determination in test set = 0.786164571607211
```

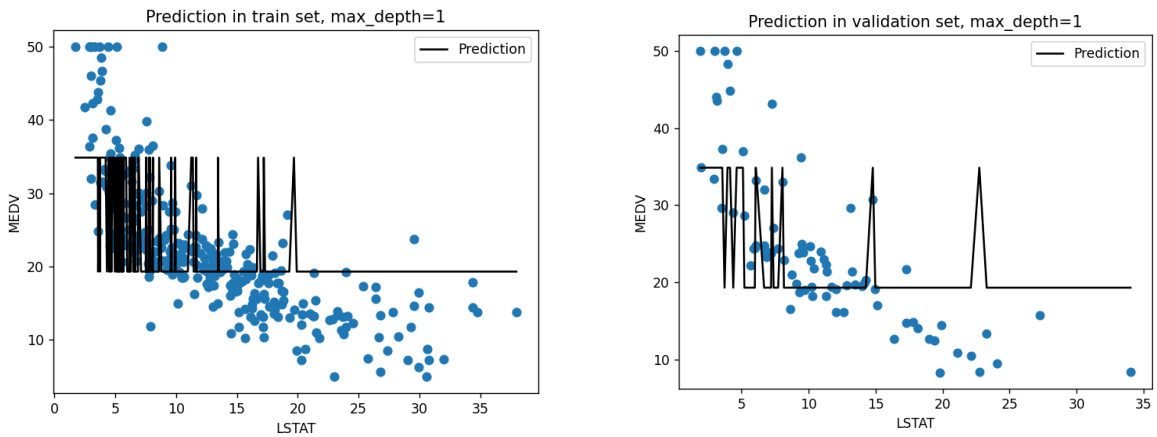
[4] Performance del modelo final en el dataset de pruebas.

```
Results from cross validation (R2)
[0.75139825 0.7304421  0.83138182 0.85909095 0.68822807]
Mean: 0.7721082379096224
Standard deviation: 0.06368705019720372
```

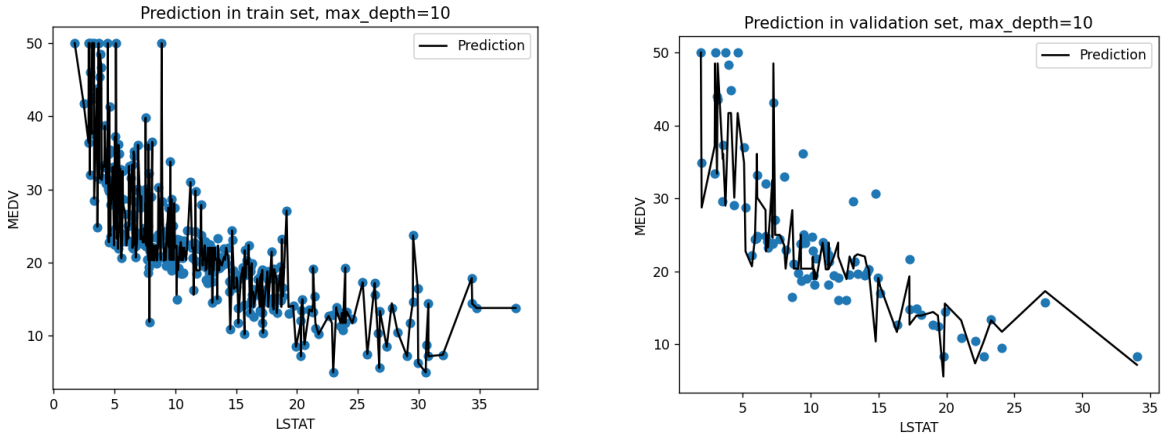
[5] Resultados de Cross-Validation para el modelo óptimo.

En la figura 4, observamos los resultados del Mean squared error con valor de 21, lo cual es más bajo que los modelos con alto max depth. Así mismo el coeficiente de determinación resultó con un valor de 0.78 en el dataset de pruebas. También se realizó cross-validation con dicho coeficiente, y se encontró que la media de esta métrica es de 0.77 y su desviación estándar de 0.06.

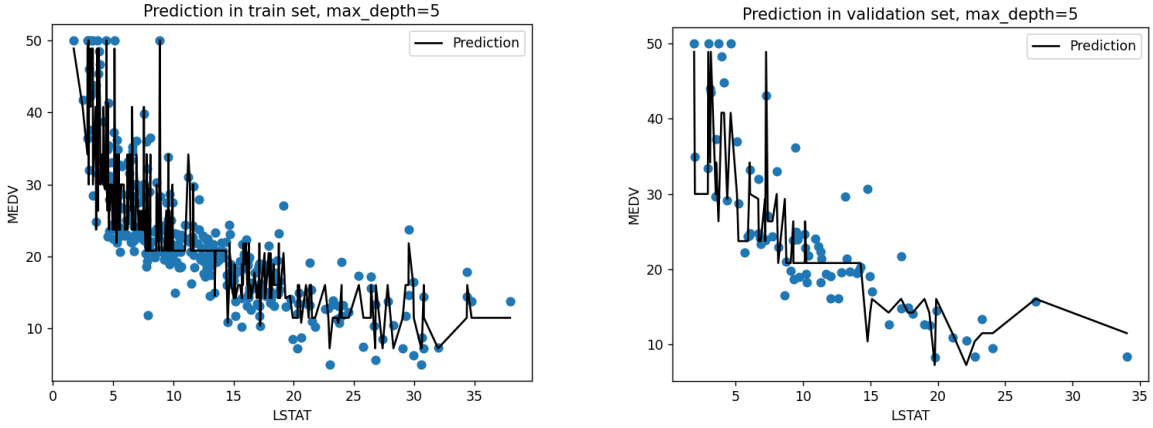
Procederemos ahora a ejemplificar cómo es que se ven tres diferentes modelos: uno con hiperparámetro que hemos identificado como underfit, el modelo overfit, y finalmente el modelo que consideramos óptimo. Para este fin, realizaremos una visualización con la variable de LSTAT, debido a su comportamiento no lineal respecto a la variable que queremos predecir (figura 1) para visualizar las diferencias de cada modelo .



[6] Modelo con hiperparámetro max depth = 1. Observamos que posee alto bias y baja varianza, ya que no captura las relaciones entre los datos y por lo tanto realiza malas predicciones tanto en el dataset de entrenamiento como en el de validación. Concluimos que el modelo sufre de underfitting.



[7] Modelo con hiperparámetro max depth = 10. El modelo posee muy bajo bias el error de la predicción en el dataset de entrenamiento es prácticamente nulo. Sin embargo, el modelo no puede predecir correctamente los datos en el dataset de validación, lo que indica alta varianza y a su vez overfitting.



[8] Modelo con hiperparámetro max depth = 5. Nos encontramos con el balance entre bias y varianza, donde ambas variables se encuentran lo más bajo posible de acuerdo a nuestro análisis previo con los scores MSE, MAE y R2. El modelo no se encuentra tan ajustado al dataset de entrenamiento como el modelo con max depth = 10, y la predicción es bastante parecida, por lo cual hemos encontrado el mejor modelo.

Conclusión

En el presente trabajo hemos presentado la influencia que tiene el sesgo (bias) y la varianza en los algoritmos de machine learning, en este caso particular, en el algoritmo de Decision Tree

Regressor de la librería de Scikit-Learn. De los tres diferentes modelos de regresión, observamos que efectivamente el que tiene mejor comportamiento es el modelo cuyo hiperparámetro no se encuentra en los extremos mínimos o máximos, y que el ajustar dicho parámetro (hypertuning), también puede considerarse como método de regularización para evitar el overfitting.

Código fuente

Disponible en https://github.com/a01752785/TC3006C-ml-algorithm-analysis.