

Clasificación de ingreso anual mayor a 50k

José Pablo Martínez Valdivia

7 de septiembre de 2024

Abstract

This paper presents a predictive model based on the Random Forest algorithm to classify income levels using the Census Income (50k) dataset. The objective of the study is to predict whether an individual's income exceeds \$50,000 per year based on demographic and work-related attributes. Furthermore, I discuss the various preprocessing decisions made, such as handling encoding categorical variables, and the choice of hyperparameters to optimize model performance.

1. Introducción

El dataset “Census Income” fue proveído por el “UCI Machine Learning Repository”. Contiene 14 variables y 48842 instancias de datos provenientes de un censo realizado en 1994 para determinar si un adulto genera un ingreso anual mayor a \$50k. Emplearemos el modelo de bosque aleatorio (random forest) para emplear una predicción categórica y disminuir el overfit en el modelo.

2.2. Transformación

Para poder general el bosque aleatorio necesitamos convertir las columnas categóricas en valores numéricos. Para esto haremos uso de *one-hot encoding*, esta técnica consta de tomar todas las categorías en una columna y crear una columna por categoría, dejando un uno en la categoría que presenta la instancia y ceros en el resto.

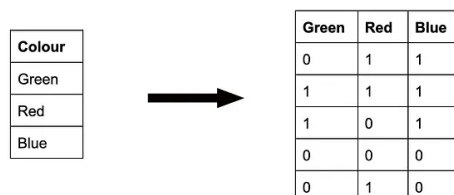


Figura 2: Método de one-hot encoding.

2. ETL

2.1. Extracción

Los datos fueron descargados por medio de la librería de “ucimlrepo”. Estos proveen datos demográficos como edad, estado civil, educación, ocupación, raza, sexo, país de origen, etc.

Los datos se encuentran completos en su mayoría, pero hay datos faltantes en las columnas de tipo de trabajo, ocupación y país de origen como se puede ver en 1. Se tomó la decisión de deshacerse de las filas con datos faltantes las cuales representan 1221 instancias, lo cual es el 2.4 % de los datos; sobrándonos 47621 instancias. Esta decisión se tomó considerando que algún tipo de imputación podría introducir errores para la tarea de clasificación.

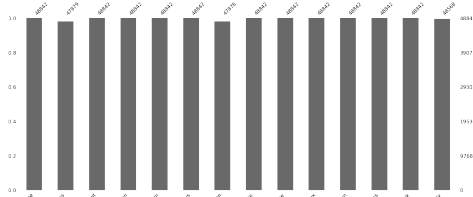


Figura 1: Valores faltantes.

3. Marco teórico

3.1. Árboles de decisión

Un árbol de decisión, específicamente para clasificación, es un modelo que nos permite producir una salida discreta. Este modelo consta de una estructura de árbol binario en la cual cada nodo padre presenta una comparación lógica de una de las entradas y las hojas representan una clase de salida.

Este modelo nos trae una serie de problemas. Para empezar, la complejidad de tiempo para obtener el resultado de un árbol es $O(n \log n)$, donde n se refiere al número de muestras con el que hayamos entrenado el árbol, por otro lado la complejidad para crear el árbol es de:

$$O(n_{muestras}n_{variables} \log n_{muestras}) \quad (1)$$

Survival of passengers on the Titanic

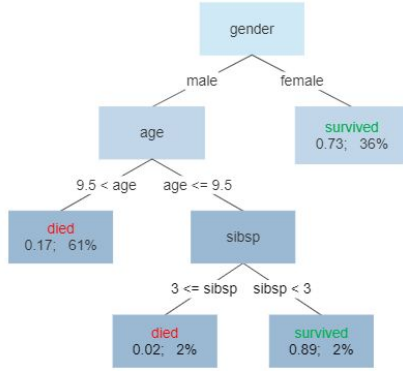


Figura 3: Estructura de un árbol de decisión.

lo cual indica que los tiempos de entrenamiento escalan $O(n \log n)$ en base a la cantidad de muestras que tengamos en el dataset. Esto quizá no presenta un problema tan grave ya que solo contamos con 48 mil entradas, pero el problema más grande de los árboles de decisión es que el generar un árbol muy complejo esta sujeto sobreajuste, es decir, que el modelo se adapte a predecir los datos de entrenamiento y por lo tanto no generalice y tenga una menor precisión al predecir datos nuevos.

3.2. Bosque Aleatorio

Una de las formas de combatir el sobreajuste inevitable de los árboles de decisión es haciendo uso de un bosque aleatorio, este es un método de ensamble en el cual, en lugar de generar un solo árbol complejo. Generamos muchos árboles pequeños y tomamos como salida la clase que haya sido más votada por todos los árboles. Este método nos permite en general mitigar el sobreajuste dada la aleatoriedad de cada árbol.

Train score: 0.833841 Test score: 0.832231

4. Entrenamiento

El set de datos fue separado en tres conjuntos: entrenamiento, validación y pruebas constanding respectivamente del 60 %, 20 % y 20 % de los datos del set. Para el modelo se decidió usar los siguientes hiperparámetros:

- n_estimators: 400
- max_leaf_nodes: 10
- max_depth: 20

Estos parámetros serán usados inicialmente para probar como se desempeña el modelo contra los datos de validación.

4.1. Entrenamiento

Las filas del dataset fueron aleatorizadas y separadas en tres conjuntos; entrenamiento, validación y prueba, estos constan del 60 %, 20 % y 20 % de los datos respectivamente. Esta separación se hizo para garantizar la eficiencia del modelo ante datos nuevos y castigar el sobre ajuste que pueda tener al trabajar solo con los datos provistos.

Posteriormente se corrió la regresión lineal con el descenso gradiente hasta que no hubiera cambio en los coeficientes o un número fijo de epochs se cumpliera. Cada epoch se calculo el MSE tomando la pérdida con el conjunto de validación y ajustando los coeficientes en magnitud del valor de aprendizaje.

5. Resultados

Después de 40,000 epochs, el modelo converge a un MSE de 0.042171 en el conjunto de validación y finalmente podemos ver las R^2 para cada conjunto en la tabla 1 comparando con el conjunto de prueba; esto representa un alto ajuste y poder predictivo del modelo con los datos reales de asesinatos en Estados Unidos. Podemos notar un grado bajo de sobreajuste para los datos de entrenamiento (1 %) al momento de comparar las R^2 de entrenamiento y validación el cual en esta magnitud no presenta un problema considerable para predecir datos nuevos. Igualmente podemos ver poca diferencia en la reducción del error 4 para los conjuntos de validación y entrenamiento. En general el modelo tiene un buen poder predictivo y no cae en sobreajuste por lo cual podemos considerar su grado de sesgo como bajo.

R^2 train	0.9554
R^2 validation	0.9335
R^2 test	0.9462

Cuadro 1: Desglose de R^2 .

La figura 4 muestra la gráfica del error de la función de pérdida a través de las epochs. Podemos ver que el modelo converge relativamente rápido y las 40000 están de más para el ajuste.

Finalmente podemos observar en la figura 5 como se comparan las predicciones con los valores reales del conjunto de prueba. Se nota inicialmente que la predicción se mantiene consistente para val-

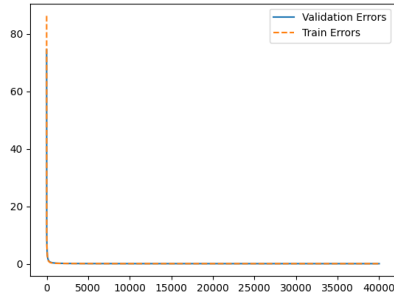


Figura 4: Función de pérdida.

ores cercanos a 0 pero comienza a divergir con valores de asesinatos más altos.

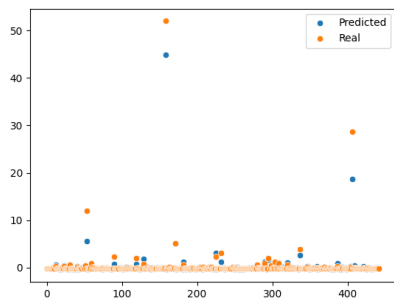


Figura 5: Resultado de la regresión.

Referencias

Redmond, Michael. (2011). Communities and Crime Unnormalized. UCI Machine Learning Repository. <https://doi.org/10.24432/C5PC8X>.