INTELIGENCIA ARTIFICIAL
ENTREGA 2 PROYECTO

PREDICCIÓN DEL NIVEL DE POBREZA EN LOS HOGARES DE COSTA RICA

PRESENTA
LINA NATALIA ANGULO DOMINGUEZ
WILLIAM MAURICIO RESTREPO VELASQUEZ

DOCENTE
RAÚL RAMOS POLLAN

FACULTAD DE INGENIERÍA
UNIVERSIDAD DE ANTIOQUIA
MEDELLÍN
2023

INTRODUCCIÓN

El banco interamericano de desarrollo está apoyando a los hogares más necesitados, para así proporcionar ayuda mediante programas de asistencia.

El objetivo es realizar una calificación (1,2,3,4) a cada hogar con el fin de identificar cuáles hogares son los que realmente necesitan asistencia. Este con el fin de mejorar el modelo de medición, el método PMT (Proxy mean test), que está basado en la observación del entorno, en cuanto al estado de la vivienda, los bienes que posee, la calidad de los materiales, los lujos, entre otros.

En el **dataset** cada fila representa la información de una persona, varias personas pueden ser parte de la misma casa, así que solo las predicciones serán para los que son cabeza de hogar.

Dataset de la competencia de Kaggle:

https://www.kaggle.com/competitions/costa-rican-household-poverty-prediction/overview

ANÁLISIS EXPLORATORIO DE LOS DATOS

Información del dataset

La variable a predecir: ***target***

1 = Pobreza extrema

2 = Pobreza moderada

3 = Hogar vulnerable

4 = Hogar no vulnerable

Este es un problema del tipo “supervisado multiclase”, ya que son 4 valores posibles los que puede tomar la variable ‘target’.

El dataset Train tiene 9557 filas y 143 columnas: 142 variables, incluyendo la columna la columna target.

El dataset Test tiene 23856 filas y 142 columnas.

Variables categóricas

El dataset contiene 5 variables categóricas, la variable ID, no aporta al modelo, ya que solo es la identificación de la persona. Entonces vamos a eliminarlo del dataframe.

Tabla 1. Dataset de las variables categóricas

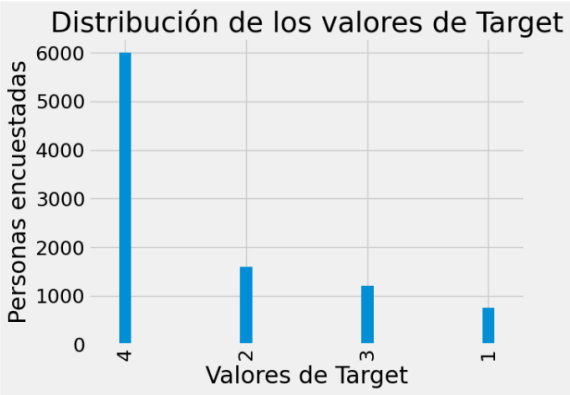
	Id	idhogar	dependency	edjefe	edjefa
0	ID_279628684	21eb7fcc1	no	10	no
1	ID_f29eb3ddd	0e5d7a658	8	12	no
2	ID_68de51c94	2c7317ea8	8	no	11
3	ID_d671db89c	2b58d945f	yes	11	no
4	ID_d56d6f5f5	2b58d945f	yes	11	no
...	...	...	...	...	...
9552	ID_d45ae367d	d6c086aa3	.25	9	no
9553	ID_c94744e07	d6c086aa3	.25	9	no
9554	ID_85fc658f8	d6c086aa3	.25	9	no
9555	ID_ced540c61	d6c086aa3	.25	9	no
9556	ID_a38c64491	d6c086aa3	.25	9	no

En este caso, se realiza la eliminación de las columnas que contienen Id e id hogar mediante un par de funciones, ya que estas no son variables de interés para realizar el modelo de aprendizaje automático, es posible que estas variables no sean relevantes para el modelo y puedan afectar negativamente el rendimiento y la precisión del modelo.

Por otro lado, se decide realizar una transformación de variables categóricas en variables numéricas, para el caso de las variables de depenency, edjefe y edjefa, para lo cual se reemplaza "no" por 0 y "sí" por 1 en variables categóricas binarias. Al hacer esta transformación, se puede tratar la variable categórica binaria como una variable numérica binaria.

Adicionalmente, se analiza el balance del dataset, realizando el conteo de las ocurrencias de la variable 'target', Si hay una categoría que tiene una proporción significativamente mayor o menor que las demás, entonces el dataset está desbalanceado y puede ser necesario tomar medidas para balancear. En este caso no se obtuvieron valores muy elevados, aunque se requiere realizar un análisis más completo y decidir si es necesario tomar medidas para balancear.

Por consiguiente se realiza un gráfico de la variable Target para así observar la distribución de los datos en cada categoría.



Gráfica 1. Distribución de los valores de Target.

VISUALIZACIÓN Y REPARACIÓN DE DATOS FALTANTES

Una vez cargamos el dataset en un dataframe de pandas,se realiza una inspección sobre las columnas que tienen valores nulos, con el fin de aplicar técnicas de llenado de la data faltante.Debido a que en el análisis de datos, trabajar con valores NaN puede llevar a resultados erróneos en los cálculos y en las visualizaciones. Además, los algoritmos de aprendizaje automático no pueden manejar valores faltantes y pueden fallar si se les presenta un conjunto de datos con valores NaN.

Por lo tanto, se procede a eliminar los valores NaN de un conjunto de datos antes de realizar el análisis o entrenar el modelo de aprendizaje automático.

La presencia de valores NaN en la variable v2a1 (pago mensual de alquiler) podría indicar que algunas de las observaciones en el conjunto de datos no tienen información sobre el pago de alquiler, o que no tienen un hogar alquilado. Esto podría deberse a varios factores, como hogares que son de propiedad propia, hogares que viven en viviendas proporcionadas por un empleador, entre otros.

Por otro lado, la variable v18q, "owns a tablet", también podría haber NaNs porque no todas las familias tienen una tablet en su hogar. Por lo tanto, para aquellos hogares que no tienen una tablet, el valor de esta variable sería NaN.

Tabla 2. Variables que contienen valores nulos.

Variable	Valor	Descripción
v2a1	6860	Pago de renta mensual

v18q1	7342	Número de tabletas que posee el hogar
rez_esc	7928	Años de atraso en la escuela
meaneduc	5	Promedio de años de educación para adultos (18+)
SQBmeaned	5	Cuadrado de la media de años de educación de los adultos (>=18) en el hogar

Para el caso del los valores de meaneduc y SQBmeaned se decidió reemplazar los NaN en por la mediana de la variable, el completar los valores faltantes por la mediana puede ayudar a mejorar la calidad del modelo predictivo, ya que permite utilizar la información de la variable en cuestión y no perder datos valiosos que podrían tener una relación importante con la variable objetivo.

Se obtiene el siguiente dataframe, luego de realizar procedimientos anteriormente mencionados.

Tabla 3. Dataset modificado de las variables categóricas.

	v2a1	v18q1	rez_esc	meaneduc	SQBmeaned
0	190000.0	0.0	0.0	10.00	100.0000
1	135000.0	1.0	0.0	12.00	144.0000
2	0.0	0.0	0.0	11.00	121.0000
3	180000.0	1.0	1.0	11.00	121.0000
4	180000.0	1.0	0.0	11.00	121.0000
...	...	...	...	...	...
9552	80000.0	0.0	0.0	8.25	68.0625
9553	80000.0	0.0	0.0	8.25	68.0625
9554	80000.0	0.0	0.0	8.25	68.0625
9555	80000.0	0.0	0.0	8.25	68.0625
9556	80000.0	0.0	0.0	8.25	68.0625

Se utilizó el clasificador gaussiano como primer enfoque para realizar la clasificación de los datos. Este es un algoritmo de clasificación supervisada que se basa en el teorema de Bayes y en la suposición de que todas las características son independientes entre sí. Esta suposición simplifica los cálculos necesarios para realizar la clasificación. Se usó debido a que es adecuado para conjuntos de datos de alta dimensionalidad, también para modelos binarios y multiclase. Además se ajustó el modelo a los datos de entrenamiento, y se obtuvo una predicción del mismo.

En consecuencia se utilizó la exactitud (accuracy), como métrica para evaluar el rendimiento del modelo, sin embargo, esta métrica puede no ser la más adecuada para evaluar el rendimiento del modelo ya que las clases están desbalanceadas.

El Accuracy inicial que se consiguió utilizando el clasificador gaussiano fue del 56.6%.

Para este tipo de modelos multiclase, se recomienda la métrica F1 score que es útil para evaluar la precisión y la exhaustividad del modelo.

El valor de F1 score que se calculó inicialmente fue de 58.5%, lo que representa un rendimiento regular.

CONCLUSIÓN

En la implementación de técnicas de Machine Learning en modelos multiclase, se pueden encontrar desafíos como el desbalanceo en las clases, la selección e interpretación de las variables relevantes.

Es importante tener en cuenta que el uso de técnicas de Machine Learning en modelos multiclase, utiliza métricas de desempeño como F1Score, las cuales ofrecen resultados más confiables que las métricas para modelos binarios.

Es necesario evaluar la dependencia de las columnas, ya que el clasificador gaussiano asume independencia entre las columnas, por lo tanto se necesitan hacer análisis de correlación, matriz de confusión, para mejorar el rendimiento del modelo.

REFERENCIAS

1. https://www.datacamp.com/tutorial/naive-bayes-scikit-learn
2. https://www.educative.io/
3. https://www.projectpro.io/article/multi-class-classification-python-example/547#mcetoc_1fpjsn4g8b