MEMORIA – Proyecto de Machine Learning

En este proyecto se utiliza el dataset Forest Cover Type procedente de la Universidad de California, Irvine, School of Information and Computer Sciences database. Se quiere predecir el tipo de cobertura forestal (cobertura terrestre formada por bosques) a partir de variables cartográficas (e.g. tipo de suelo, elevación, pendiente, sombra, distancia a ríos, distancia a focos de incendios forestales).

Introducción a los datos

El set de datos utilizado se llama Forest Cover Type Data Set y se descarga en el siguiente enlace: https://archive.ics.uci.edu/ml/datasets/Covertype. Los datos se obtienen de la web University of California Machine Learning Repository https://archive.ics.uci.edu/ml/datasets.php. La fuente original de los datos y sus propietarios son: Remote Sensing and GIS Program, Department of Forest Sciences. College of Natural Resources. Colorado State University. Fort Collins, CO 80523.

El archivo se descarga en formato .data (covtype.data) junto con un archivo en formato .info (covtype.info) que contiene información del dataset.

El número de observaciones es de 581,012. El número de atributos son 54 columnas de datos (10 variables cuantitativas, 4 variables binarias referentes a cuatro áreas de estudio diferentes, y 40 variables binarias del tipo de suelo). La descripción de las variables viene en la Tabla 1.

La variable a predecir es una clase del tipo de cobertura forestal. Son 7 clases:

1. Conífera/Abeto (Spruce/Fir)

2. Pino Lodgepole (Lodgepole Pine)

3. Pino Ponderosa (Ponderosa Pine)

4. Álamo/Sauce (Cottonwood/Willow)

5. Álamo temblón (Aspen)

6. Abeto de Douglas (Douglas-fir)

7. Krummholz (tipo de vegetación atrofiada y deformada)

Tabla 1: Descripción de las variables (Blackard, 1998)

Table 1 Data attribute information

Attribute name	Measurement unit	Attribute description
Elevation	Meters	Elevation
Aspect	Azimuth	Aspect in degrees
Slope	Degrees	Slope
Horizontal distance to hydrology	Meters	Horizontal distance to nearest surface water features
Vertical distance to hydrology	Meters	Vertical distance to nearest surface water features
Horizontal distance to roadways	Meters	Horizontal distance to nearest roadway
Hill shade at 9 am	0–255 (Index)	Hill shade index at 9am, summer solstice
Hill shade at noon	0–255 (index)	Hill shade index at noon, summer solstice
Hill shade at 3 pm	0–255 (Index)	Hill shade index at 3pm, summer solstice
Horizontal distance to fire points	Meters	Horizontal distance to nearest wildfire ignition points
Wilderness area (4 binary columns)	0 (Absence) 1 (Presence)	Wilderness area designation
Soil type (40 binary columns)	0 (Absence) 1 (Presence)	Type of soil
Forest cover type (7 types)	1–7	Type of forest cover

La zona de estudio es el Bosque Nacional Roosevelt al norte de Colorado. Esta zona incluye cuatro áreas silvestres (‘Wilderness_Area’):

1. Rawah (Area 1)
2. Neota (Area 2)
3. Comanche Peak (Area 3)
4. Cache la Poudre (Area 4)

¿Para qué predecir la cobertura forestal?

Los administradores de recursos naturales (natural resource managers, e.g. Ministerio de Medio Ambiente, Parques Nacionales, Ayuntamientos, Organizaciones de Conservación de la Naturaleza) requieren información descriptiva básica, incluidos datos de inventario de coberturas forestales, para respaldar sus procesos de toma de decisiones. Sin embargo, los administradores generalmente no cuentan con este tipo de datos para los terrenos que

se encuentran fuera de su jurisdicción inmediata. Un método para obtener esta información es mediante el uso de modelos predictivos.

Exploratory Data Analysis

El set de datos viene sin missing values. Existen outliers en la variable 'Vertical Distance to Hydrology' que son valores negativos que no tienen sentido al tratarse de metros. Hay varias columnas que presentan correlación como las 'Hillshade' o la 'Vertical Distance to Hydrology' con la 'Horizontal Distance to Hydrology'. El target no está balanceado, hay muchos más valores de las clases 1 y 2 que del resto. Se lleva a cabo un RandomUnderSampler y SMOTE() para equilibrar el target. El SMOTE() tarda cerca de unos 20 minutos en hacerse, creo a que duplica las filas del dataset original, a cerca de 1,800.000.

Preprocesado de los datos

Se hace una separación de los datos en train y test (80 %, 20 % de los datos, respectivamente). Después los datos se estandarizan con un StandardScaker.

Feature Selection

Se aplican métodos de Feature Selection para obtener la máxima información con el mínimo uso de recursos. Encontrar un modelo sencillo con menos variables que tenga un alto poder explicativo.

Primero se aplica un Random Forest Selection. Como criterio general se excluyen las variables que tiene la 'Variable Importance' (VI) más baja. La regla general, sin embargo, es eliminar las variables que tienen una proporción de importancia inferior al 5 %, pero dado que solo tenemos 12 variables (o 54 atributos, debido a las variables categóricas), excluir 5 (VI por debajo del 5 %) de las 12 variables podría no ser una buena elección. Por lo tanto, establecemos nuestro propio criterio para excluir las variables que tienen el valor de VI más pequeño de todas las variables, que en nuestro caso es la variable 'Slope', pendiente.

Segundo, se aplica una PCA. En este caso no hay una respuesta clara sobre dónde se encuentra el límite de la varianza. Sin embargo, teniendo en cuenta que el número de componentes principales (PC) es de 54, muchos de ellos contribuyen de manera insignificante a la varianza proporcional. Se decide poner el umbral en el 80% de la varianza proporcional, lo que selecciona los primeros 33 componentes principales.

Algoritmos

Primero, se aplica un RandomForestClassifier y da un accuracy de 0.95, parece un buen clasificador.

Después se aplica una Logistic Regression y da una accuracy de 0.71. Al hacer la reducción de dimensiones de la PCA no mejora el score, se queda igual. Y tampoco al reducir las dimensiones en base al Random Forest Selection. Esto da información de que, aunque las predicciones no mejoren ni empeoren, al reducir las dimensiones y, por tanto, simplificar el modelo y reducir el gasto de computación, se obtiene un valor igual. Con menos recursos, igual predicción. Esto puede ser importante cuando tengamos set de millones de datos para ahorra tiempo.

Volviendo al RandomForestClassifier de accuracy de 0.95, se quiere intentar mejorar aún más. Para ello se hace un GridSearchCV con diferentes parámetros y se usan los datos de entrada con el target balanceado mediante RandomUnderSampler. El mejor módelo es un RandomForestClassifier (max_features = 6, n_estimators = 600) que devuelve un accuracy de 0.87. Mejor quedarse con la versión anterior que tiene un scoring más alto.

Después se aplica un Decission Tree Classifier con los datos originales del dataset de train y test estandarizados. El mejor modelo es un DecisionTreeClassifier(max_depth = 7, min_samples_leaf = 1) con un accuracy de 0.73.

Por último, se aplica un KNN con los datos de train y test resultantes del RandomUnderSampler. Da un accuracy de 0.78.

Los pasos detallados vienen en el Notebook llamado main.ipynb con las celdas y los comentarios.

Estructura de carpetas

src/: archivo main.ipynb (notebook con el detalle de todos los pasos)
src/data/raw: descargados de la web University of California Machine Learning Repository.
src/visualizaciones: plots creados en el notebook

Librerías utilizadas

- numpy as np
- pandas as pd
- matplotlib.pyplot as plt
- seaborn as sns
- sklearn

Técnicas utilizadas

1. EDA: manejar bases de datos, filtrarlas, añadir y eliminar columnas y filas, localizar datos, eliminar outliers, agrupar por columnas, describir la estadística general del dataframe, comprobar balanceo de la variable a predecir.
2. Feature Engineering. En concreto, Feature Selection mediante Random Forest Selection y PCA.
3. Preprocesado de datos: división de train y test; estandarización, escalado
4. Técnicas de mejora para escoger el mejor modelo: GridSearchCV, prueba de diferentes parámetros del modelo, prueba de dos maneras de balancear la target (RandomUnderSampler y SMOTE)
5. Entrenar modelos para hacer predicciones de un clasificador. En concreto, el Random Forest Classification, Logistic Regression, Decision Tree Classifier y KNN. Uso de métricas para comparar.

Consideraciones

Se ha intentado aplicar otros modelos como SVM y XGBoost, hacer más GridSearchCV pero no se han podido hacer por falta de recurso computacional ya que una vez tardaba más de 40 min en ejecutar la celda, se ha interrumpido el proceso voluntariamente por bloqueo del ordenador. Por otro lado, a fututo sería bueno probar con una red neuronal.

Referencias:

- Blackard, J. A., & Dean, D. J. (1999). Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, *24*(3), 131-151.
- Kumar, A., & Sinha, N. (2020). Classification of forest cover type using random forests algorithm. In *Advances in data and information sciences* (pp. 395-402). Springer, Singapore.
- Sjöqvist, H., Längkvist, M., & Javed, F. (2020). An analysis of fast learning methods for classifying forest cover types. *Applied Artificial Intelligence*, *34*(10), 691-709.