Tecnológico de Monterrey

Momento de Retroalimentación: Módulo 2

Random Forest

Materia:

Inteligencia artificial avanzada para la ciencia de datos I

(Gpo 101)

Alumno:

Alfredo Azamar López - A01798100

Profesor:

Jorge Adolfo Ramírez Uresti

Tec de Monterrey Septiembre 2024

Para este entregable se utilizará el algoritmo supervisado de *Random Forest*, algoritmo que se caracteriza por versatilidad (para tareas de regresión como clasificación) y su robustez contra el sobreajuste.

El dataset seleccionado está relacionado con el cáncer de mama, la información proviene de la librería de sklearn, para este contexto es útil utilizar *Random Forest* debido a la importancia de maximizar la precisión al detectar correctamente los casos malignos o benignos; al igual que la naturaleza no lineal de las características del dataset.

Hiper Parámetros

Para nuestro modelo se utilizaron los siguientes hiper parámetros:

- `n_estimators`

Nos indica el número de árboles que se crean en nuestro bosque, generalmente aumentamos el número de árboles para mejorar la precisión del modelo (Aunque este incrementa el tiempo de entrenamiento).
Ejemplo de uso: `'n_estimators': [50, 100, 200]`

- `max_depth`

Es la profundidad máxima que tiene cada árbol, cuando se limita esa profundidad ayudamos a controlar el sobreajuste, un valor bajo de profundidad evita que se ajuste demasiado a los datos (hay que recordar que los valores altos de profundidad nos permiten modelar nuestros sistemas más complejos).
Ejemplo de uso: `'max_depth': [2, 10, 20, 30]`

- **`min_samples_split`**

Representa el número mínimo de muestras para dividir un nodo, cuando le proporcionamos un número alto este evita la decisión excesiva de los árboles (reduce la variabilidad).
Ejemplo de uso: `'min_samples_split': [2, 5, 10]`

- **`min_samples_leaf`**

Este parámetro nos indica el número de muestras que debe de tener una hoja, este controla el tamaño mínimo de los nodos terminales, previene que el modelo cree nodos hojas muy pequeñas (crea un modelo muy generalizado)
Ejemplo de uso: `'min_samples_leaf': [1, 2, 4]`

Para optimizar estos hiper parámetros se utilizó *GridSearchCV*, esta herramienta nos ayuda a encontrar un punto medio entre la capacidad de modelo para generar datos y su precisión en el conjunto de entrenamiento, garantizándonos un modelo equilibrado y eficiente.

Ejemplo de uso:

```
rfBase = RandomForestClassifier(random_state=42)
gridSearch = GridSearchCV(rfBase, param_grid, cv=5) # 5-fold cross-validation
```

Resultados

En la consola obtenemos datos como los mejores parámetros y estimadores que se encontraron para el algoritmo, también se imprime la proyección de la variable así como la exactitud del modelo y se calcula y gráfica una matriz de confusión.

```
Mejores parámetros: {'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}

Mejor estimador: RandomForestClassifier(max_depth=10, min_samples_leaf=2, n_estimators=200,
                       random_state=42)

Exactitud: 0.9707602339181286

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.98      0.94      0.96        63
           1       0.96      0.99      0.98       108

    accuracy                           0.97       171
   macro avg       0.97      0.96      0.97       171
weighted avg       0.97      0.97      0.97       171
```

Matriz de confusión: