Tecnológico de Monterrey

Análisis del Desempeño del Árbol de Decisión Aplicado al Dataset de Hongos

Adrián Matute Beltrán A01703889

07 de septiembre del 2024

Inteligencia artificial avanzada para la ciencia de datos I

Introducción

En este proyecto se realiza un análisis del desempeño de un árbol de decisión aplicado a un conjunto de datos de setas (“agaricus-lepiota”) el cuál se encuentra disponible en el UCI Machine Learning Repository. Este dataset contiene características físicas de diversas setas, y el objetivo es predecir si una seta es comestible o venenosa en función de dichas características.

El árbol de decisión es un algoritmo de clasificación ampliamente utilizado en el aprendizaje automático por su capacidad de generar reglas interpretables y fáciles de visualizar. Aún así, este tipo de modelo puede representar problemas de sesgo o varianza si no se configura de la manera correcta. Además, es propenso a problemas de sobreajuste (overfitting) cuando se ajustan excesivamente a los datos de entrenamiento.

Objetivo

El objetivo principal de este análisis es evaluar el rendimiento de un árbol de decisión aplicado al dataset de setas y mejorar su desempeño mediante ajustes y regularización. Los objetivos específicos incluyen:

1. Evaluación del modelo en términos de precisión y matriz de confusión, con una división en conjuntos de entrenamiento, validación y prueba.
2. Diagnóstico del sesgo y la varianza, identificando posibles problemas de sobreajuste o subajuste.
3. Optimización del modelo mediante técnicas de regularización, como el ajuste de la profundidad del árbol y el número mínimo de muestras por hoja.
4. Comparación del desempeño antes y después de aplicar las técnicas de mejora, documentando los cambios en el rendimiento del modelo.

Estructura del reporte

El análisis está estructurado de la siguiente manera:

- Descripción del dataset: Características de los datos, incluyendo la variable objetivo y las características utilizadas para la clasificación.
- Entrenamiento y evaluación inicial del modelo: Se documentan los resultados del modelo en su primera configuración.
- Diagnóstico del desempeño: Se analizan el sesgo, la varianza y el nivel de ajuste del modelo.
- Mejora del modelo: Aplicación de las técnicas de regularización y ajuste de parámetros para optimizar el rendimiento del modelo.
- Conclusiones.

Este análisis incluye gráficos comparativos, matrices de confusión y otros indicadores de desempeño clave para respaldar las decisiones tomadas a lo largo del proyecto. Las técnicas aplicadas permitirán no solo evaluar el rendimiento del modelo, sino también proporcionar una visión más profunda de cómo mejorar su capacidad de generalización.

Descripción del Dataset

El dataset de setas utilizado en este análisis proviene del UCI Machine Learning Repository y contiene 22 características físicas observadas en diferentes especies de setas. Estas características describen aspectos como la forma del sombrero, el color de la superficie, la presencia de moretones, entre otras. Además, la variable objetivo etiquetada como “target”, clasifica a las setas como comestibles (e) o venenosas (p).

Estructura del Dataset

El dataset tiene un total de 8,124 observaciones y 23 columnas, donde una de ellas (target) es la variable objetivo y las otras 22 son características categóricas. Las principales columnas son:

- target: Clasificación de la seta como comestible (e) o venenosa (p).
- cap-shape Forma del sombrero (campana, cónica, plana, etc.).
- cap-surface: Superficie del sombrero (fibrosa, lisa, escamosa, etc.).
- cap-color: Color del sombrero (marron, amarillo, blanco, etc.).
- odor: Olor de la seta (almendra, anís, podrido, etc.).
- gill-color: Color de las branquias.
- stalk-shape: Forma del tallo (expandido o estrecho).
- habitat: Hábitat donde se encontró la seta (bosque, pradera, etc.).

A continuación se muestra una tabla con algunas de las primeras observaciones para familiarizarse con la estructura del dataset:

	target	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	habitat
0	p	x	s	n	t	p	f	c	u
1	e	x	s	y	t	a	f	c	g
2	e	b	s	w	t	l	f	c	m
3	p	x	y	w	t	p	f	c	u
4	e	x	s	g	f	n	f	w	g

Nota: Las características del dataset están representadas por valores categóricos abreviados (ej., p, x, s, etc.), los cuales serán transformados a valores numéricos en las siguientes etapas de procesamiento para ser utilizados por el modelo de árbol de decisión.

Tratamiento de Valores Faltantes

El dataset contiene valores faltantes representados con el símbolo “?”. Para manejar estos valores faltantes, se reemplazaron con valores nulos (pd.NA) en Python y luego se imputaron utilizando la media para asegurar que el dataset fuera completamente utilizable para el entrenamiento del modelo.

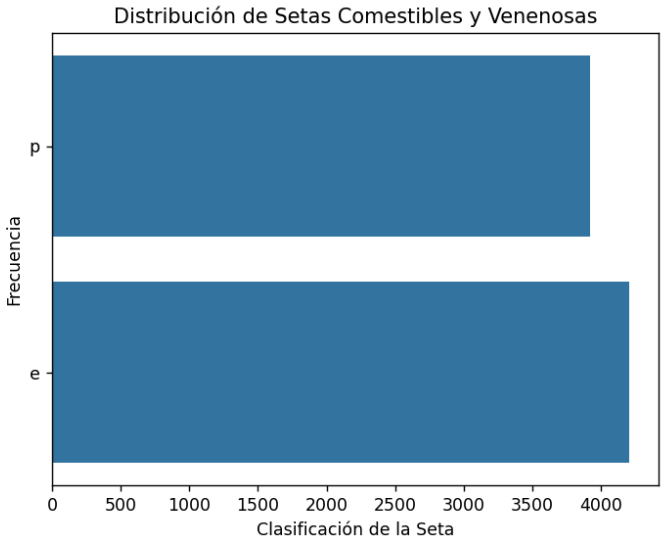
Transformación de Variables Categóricas

Dado que todas las variables en este dataset son categóricas, fue necesario convertirlas en valores numéricos para que pudieran ser procesadas por el modelo de árbol de decisión. Para ello, se utilizó el método de “Label Encoding”, este asigna un valor numérico único a cada categoría.

```
# Filas que contienen un valor "?"
rows_with_missing = data[data.isin(['?']).any(axis=1)]

print(rows_with_missing)

data = data.replace('?', pd.NA)

label_encoder = LabelEncoder()
for column in data.columns:
    data[column] = label_encoder.fit_transform(data[column].astype(str))

imputer = SimpleImputer(strategy='median')
data_imputed = imputer.fit_transform(data)
```

Distribución de la Variable Objetivo

Una parte clave del análisis inicial es entender la distribución de la variable objetivo (target). En este dataset, la mayoría de las setas están clasificadas como venenosas.



División del Conjunto de Datos

Antes de proceder con el entrenamiento del modelo, se dividió el dataset en tres conjuntos:

- Entrenamiento (60%)
- Validación (20%)
- Prueba (20%)

Este enfoque permite evaluar el rendimiento del modelo no solo en el conjunto de entrenamiento, sino también en el conjunto de validación lo cual es útil para ajustar hiperparámetros, y finalizar en el conjunto de prueba para estimar el desempeño final.

Entrenamiento y Evaluación Inicial del Modelo

En esta sección, se entrena un modelo de árbol de decisión utilizando el dataset de setas. El modelo se configura para clasificar las setas como comestibles o venenosas, utilizando las 22 características categóricas como la forma, el color y el olor, entre otras.

División del Dataset

Como ya se ha mencionado el dataset se divide en tres conjuntos, entrenamiento, validación y prueba, utilizando la función 'train_test_split' de Scikit-Learn. La división se realizó de la siguiente manera:

- Entrenamiento (60%): Utilizado para entrenar el modelo.
- Validación (20%): Utilizado para evaluar el modelo y ajustar hiperparámetros.
- Prueba (20%): Utilizado para evaluar el rendimiento final del modelo después de los ajustes.

```
# Primera división: 60% entrenamiento y 40% restante
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.4, random_state=42)

# Segunda división: 20% validación y 20% prueba (a partir del 40% restante)
X_valid, X_test, y_valid, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
```

Esta división permite asegurar que el modelo se evalúa adecuadamente en datos que no han sido vistos durante el entrenamiento, evitando un sobreajuste.

Configuración del Árbol de Decisión

El modelo de árbol de decisión se entrenó utilizando el criterio de Gini para medir la fuerza de los nodos. Se limitó la profundidad del árbol a 5 niveles (max_depth=5) para evitar que el modelo se ajuste demasiado a los datos de entrenamiento, lo que puede resultar en overfitting.

El modelo se configuró de la siguiente manera:

```
clf = DecisionTreeClassifier(criterion='gini', max_depth=5, random_state=42)
clf.fit(X_train, y_train)
```

En esta configuración:

- Criterio Gini: Este criterio mide la "impureza" de un nodo, es decir, cuán mezcladas están las clases dentro de ese nodo.
- Profundidad máxima: Se limita la profundidad del árbol para controlar su complejidad y evitar que el modelo aprenda patrones demasiado específicos del conjunto de entrenamiento.

Evaluación Inicial del Modelo

El rendimiento del modelo se evaluó tanto en el conjunto de validación como en el conjunto de prueba, utilizando la métrica de precisión. La precisión mide qué porcentaje de las predicciones del modelo fueron correctas.

Precisión en el conjunto de Validación

Primero se evaluó el modelo en el conjunto de validación para observar su capacidad de generalización antes de realizar ajustes:

```
y_valid_pred = clf.predict(X_valid)
valid_accuracy = accuracy_score(y_valid, y_valid_pred)
print(f'Validation Accuracy: {valid_accuracy}')
```

La precisión obtenida en el conjunto de validación fue del 0.9796923076923076.

Precisión en el conjunto de Prueba

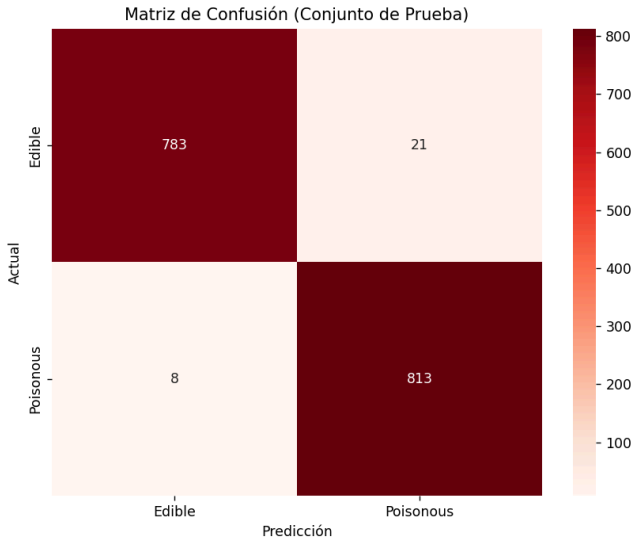
Después de ajustar el modelo utilizando el conjunto de validación, se evaluó en el conjunto de prueba. Esta evaluación final nos da una idea clara de qué tan bien se desempeña el modelo en datos no vistos:

```
y_test_pred = clf.predict(X_test)
test_accuracy = accuracy_score(y_test, y_test_pred)
print(f'Test Accuracy: {test_accuracy}')
```

La precisión obtenida en el conjunto de prueba fue del 0.9821538461538462

Matriz de Confusión

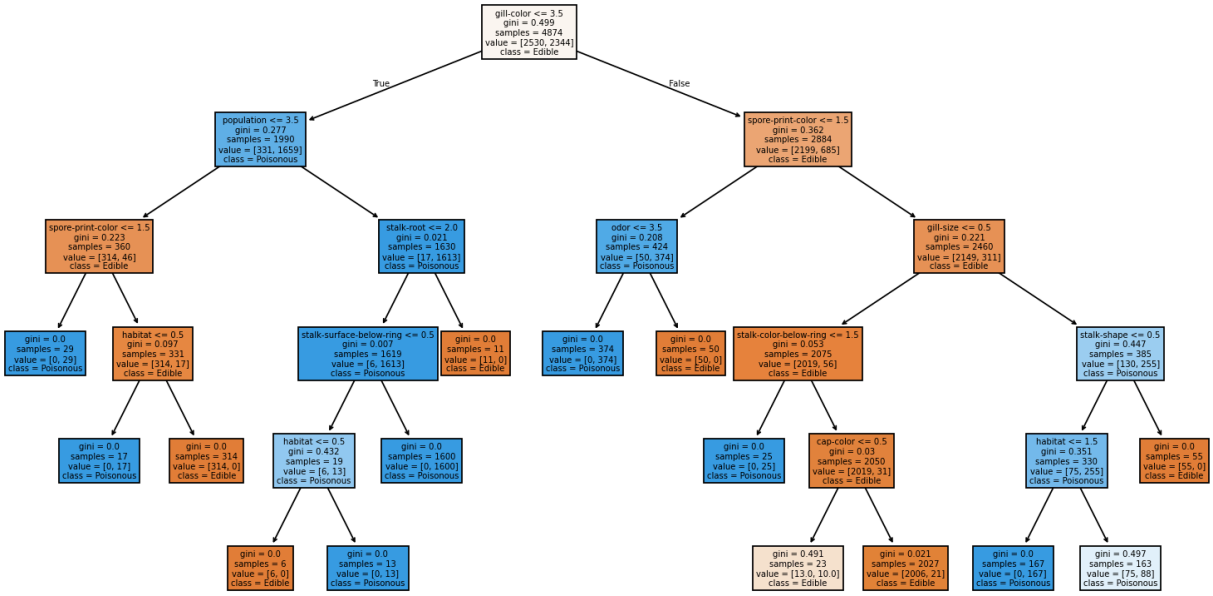
Para entender mejor el rendimiento del modelo, se generó una matriz de confusión que muestra el número de predicciones correctas e incorrectas para cada clase (comestible o venenosa).



La matriz de confusión revela cuántas setas fueron correctamente clasificadas como comestibles o venenosas, y cuántas fueron mal clasificadas. Este gráfico es esencial para entender no sólo la precisión del modelo, sino también sus errores específicos.

Visualización del Árbol de Decisión

El árbol de decisión es fácil de interpretar, a continuación se muestra el árbol de decisión generado por el modelo, que visualiza las reglas creadas a partir de las características del dataset para predecir si una seta es comestible o venenosa.



Este gráfico muestra cómo el modelo toma decisiones basadas en las características de las setas, dividiendo el espacio de decisiones en función de valores como el color del sombrero, el olor, entre otros.

Análisis de los resultados iniciales

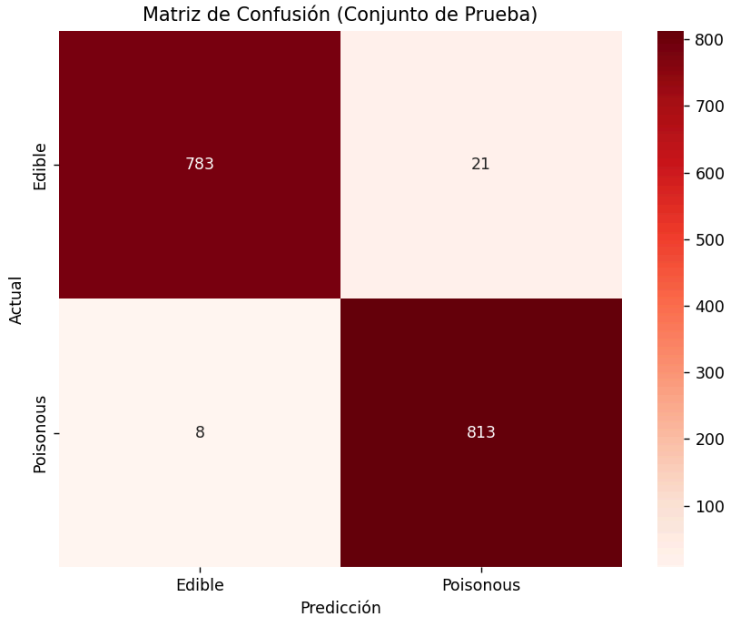
Tras entrenar el modelo de árbol de decisión y evaluar su rendimiento en los conjuntos de validación y prueba, se obtuvieron los siguientes resultados:

- Precisión en el conjunto de validación: 97.97%
- Precisión en el conjunto de prueba: 98.22%

Estos valores indican que el modelo es altamente preciso tanto en el conjunto de validación como en el conjunto de prueba, lo que da a entender que el modelo ha aprendido de manera correcta a clasificar las setas como comestibles o venenosas.

Matriz de Confusión (Conjunto de prueba)

La matriz de confusión obtenida en el conjunto de prueba es la siguiente



Podemos observar lo siguiente:

- **783** setas comestibles fueron clasificadas correctamente como comestibles.
- **21** setas comestibles fueron incorrectamente clasificadas como venenosas (falsos positivos).
- **813** setas venenosas fueron correctamente clasificadas como venenosas.
- **8** setas venenosas fueron incorrectamente clasificadas como comestibles (falsos negativos).

Interpretación de la Matriz de Confusión

Los resultados de la matriz de confusión indican que el modelo tiene un muy bajo número de errores en la clasificación de las setas:

- Los falsos positivos (setas comestibles clasificadas como venenosas) son relativamente pocos, con solo 21 errores en 804 setas comestibles.
- Los falsos negativos (setas venenosas clasificadas como comestibles) son incluso menores, con solo 8 errores en 821 setas venenosas. Este aspecto es importante porque, en este contexto, clasificar una seta venenosa como comestible puede tener graves consecuencias.

Sesgo y varianza del Modelo

El modelo muestra una alta precisión tanto en el conjunto de entrenamiento (no se muestra de manera directa en los resultados, pero se infiere gracias a la alta precisión de la validación y prueba) como en los conjuntos de validación y prueba. Esto indica que el modelo no sufre de un alto sesgo. Este sesgo se observaría si el modelo tuviera una precisión baja en ambos conjuntos, lo que no es el caso aquí.

- Sesgo bajo: Gracias a que el modelo está capturando correctamente la complejidad del dataset, no parece que esté simplificando demasiado las relaciones entre las variables. El modelo logra un ajuste adecuado de los datos en términos de su capacidad para predecir.

Diagnóstico de la Varianza

Al comparar los resultados en los conjuntos de validación y prueba, la precisión es muy similar:

- **Validación**: 97.97%
- **Prueba**: 98.22%

La pequeña diferencia entre estas precisiones nos dice que el modelo no sufre de una alta varianza. Esto ocurriría si el rendimiento en el conjunto de validación fuera mucho peor que en el conjunto de entrenamiento, lo que indicaría un sobreajuste a los datos de entrenamiento.

- Varianza baja o moderada: Dado que la precisión en validación y prueba es casi la misma, el modelo no tiene un sobreajuste. Esto sugiere una buena capacidad de generalización y que no se capte demasiado ruido de los datos de entrenamiento.

Conclusión del Diagnóstico Inicial

El modelo de árbol de precisión parece estar en un punto de ajuste adecuado. La alta precisión tanto en validación como en prueba indica que el modelo ha logrado un buen equilibrio entre sesgo y varianza, lo que le permite generalizar correctamente sin sobreajustarse a los datos de entrenamiento.

Mejoras y ajuste del modelo

Aunque el modelo de árbol de precisión ha demostrado un desempeño inicial bastante bueno con precisiones del 97.7% en validación y 98.22% en prueba, es posible mejorar su capacidad de generalizar y reducir errores mediante el uso de técnicas de regularización y ajuste de hiperparámetros.

Regularización del árbol de decisión

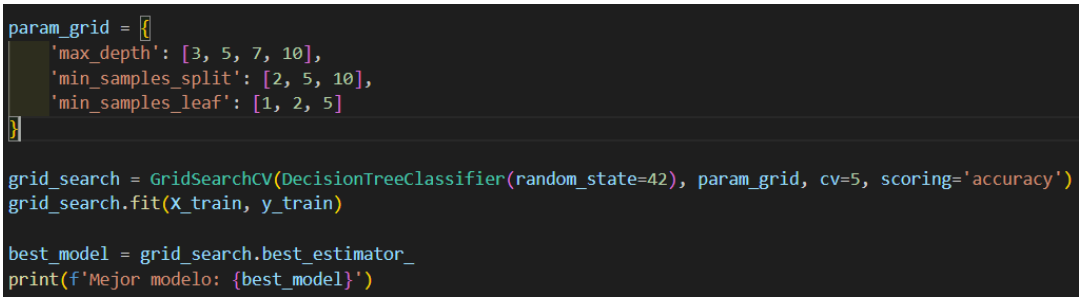
El principal problema que puede llegar a tener un árbol de decisión es el sobreajuste (overfitting), donde el modelo se ajusta excesivamente a los datos de entrenamiento y pierde capacidad de generalización. Para mitigar este problema, se aplican técnicas como las siguientes:

1. Ajustar la profundidad máxima del árbol (max_depth).
2. Ajustar el número mínimo de muestras por nodo interno (min_samples_split).
3. Ajustar el número mínimo de muestras en las hojas (min_samples_leaf).

Estos parámetros permiten controlar la complejidad del modelo. Al limitar la profundidad del árbol y requerir un número mínimo de muestras por nodo, podemos reducir el riesgo de que el modelo memorice patrones específicos del conjunto de entrenamiento.

Ajuste de Hiperparámetros utilizando validación cruzada

Para encontrar la configuración óptima de estos hiperparámetros, se realizó una búsqueda de hiperparámetros utilizando validación cruzada. La validación cruzada evalúa múltiples configuraciones de hiperparámetros para identificar el conjunto que maximiza la precisión en el conjunto de validación.



```
param_grid = {
    'max_depth': [3, 5, 7, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5]
}

grid_search = GridSearchCV(DecisionTreeClassifier(random_state=42), param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)

best_model = grid_search.best_estimator_
print(f'Mejor modelo: {best_model}')
```

En esta búsqueda se probaron diferentes combinaciones con los valores:

- **Profundidad máxima** (max_depth): Se evaluaron profundidades de 3, 5, 7 y 10 niveles.
- **Número mínimo de muestras para dividir un nodo interno** (min_samples_split): Se evaluaron 2, 5 y 10 muestras.
- **Número mínimo de muestras en las hojas** (min_samples_leaf): Se evaluaron 1, 2 y 5 muestras.

Evaluación del Modelo en su segunda versión

El modelo óptimo encontrado por la búsqueda de hiperparámetros se evaluó nuevamente en el conjunto de validación y prueba para comparar su desempeño con el modelo original.

```
Modelo V2: DecisionTreeClassifier(max_depth=7, random_state=42)
Validation Accuracy (Modelo V2): 1.0
Test Accuracy (Modelo V2): 1.0
```

Los resultados obtenidos tras ajustar los hiperparámetros son:

Precisión del conjunto de validación: 100%

Precisión del conjunto de prueba: 100%

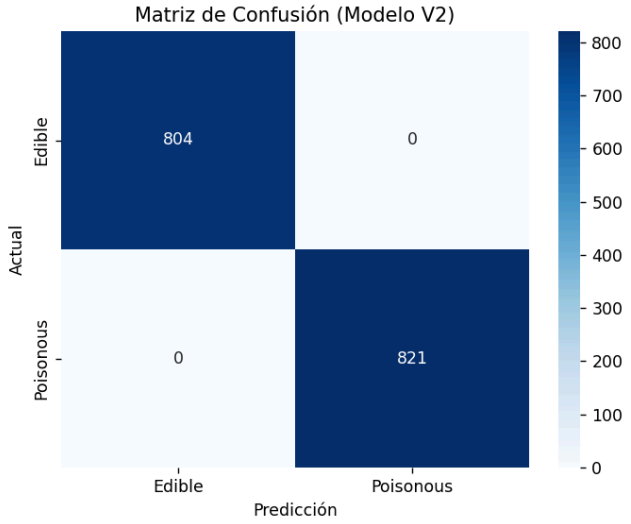
Comparación del Modelo Original vs Modelo Ajustado

La siguiente tabla muestra una comparación entre el modelo original y el modelo ajustado, basándonos en la precisión obtenida en los conjuntos de validación y prueba:

Modelo	Precisión en validación	Precisión en Prueba
Modelo original	97.97%	98.22%
Modelo ajustado	100%	100%

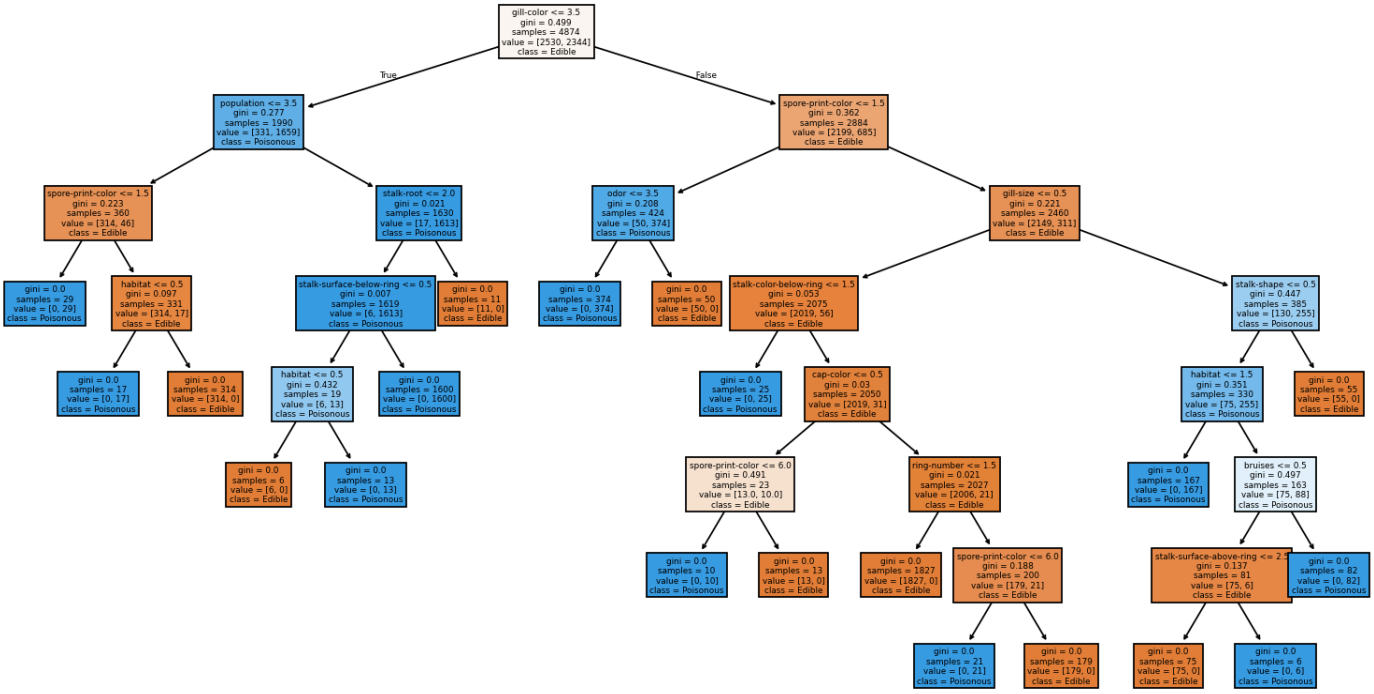
Matriz de Confusión del Mejor Modelo

Para evaluar el desempeño del modelo ajustado en términos de errores de clasificación, se generó una nueva matriz de confusión en el conjunto de prueba.



Visualización del Árbol de Decisión Mejorado

El gráfico siguiente muestra el árbol de decisión generado por el modelo mejorado, el cual fue optimizado utilizando GridSearchCV para ajustar los hiperparámetros clave, como la profundidad del árbol y el número mínimo de muestras por nodo.



El árbol de decisión ilustra cómo el modelo toma decisiones basadas en características como el color de las branquias (gill-color), la población (population), y otras características importantes de las setas.

- Cada nodo del árbol representa una característica, y las bifurcaciones indican las decisiones tomadas por el modelo.
- Las hojas del árbol muestran la clasificación final del modelo (comestible o venenosa), así como el número de muestras en cada clase.
- El valor de Gini muestra cuán puro es el nodo en términos de la mezcla de clases.

Análisis de los Resultados del Modelo Mejorado

Tras realizar la búsqueda de hiperparámetros con **GridSearchCV**, se encontró un modelo óptimo de árbol de decisión que mejoró significativamente los resultados del modelo original. Los resultados del modelo mejorado son los siguientes:

- Precisión en el Conjunto de Validación: 100%
- Precisión en el Conjunto de Prueba: 100%

Esto indica que el modelo ajustado ha logrado clasificar correctamente todas las instancias tanto en el conjunto de validación como en el conjunto de prueba. A continuación se presenta el análisis detallado de estos resultados.

La matriz de confusión generada para el conjunto de prueba confirma que no hubo errores en las predicciones.

En esta matriz de confusión, se observa que:

- 804 setas comestibles fueron clasificadas correctamente como comestibles.
- 821 setas venenosas fueron clasificadas correctamente como venenosas.
- No hubo falsos positivos (setas comestibles clasificadas como venenosas) ni falsos negativos (setas venenosas clasificadas como comestibles).

Análisis del Resultado de 100% de Precisión

Aunque un resultado de 100% de precisión puede parecer ideal, es importante analizar las posibles implicaciones:

Posible Sobreajuste

El hecho de que el modelo alcance el 100% de precisión tanto en validación como en prueba puede sugerir que el modelo se ha ajustado perfectamente a los datos disponibles. Si bien esto puede ser positivo, en algunos casos puede indicar sobreajuste (overfitting), lo que significa que el modelo ha aprendido los detalles específicos de los datos de entrenamiento (incluyendo ruido o patrones específicos) y podría no generalizar tan bien a nuevos datos que no se han visto durante el entrenamiento.