Clasificación de hongos

Gabriel Eduardo Camargo García
Francisco Javier Silva Cadavid

Asignatura
Deep Learning
Docente
Raúl Ramos Pollán

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Universidad de Antioquia
Medellín
2023

I. Estructura del Notebook

El notebook está estructurado de la siguiente manera:

1. Librerías, lectura y ajustes del data set: En esta sección se importan las librerías necesarias para cargar, entrenar y ajustar el data set. Se lleva a cabo la lectura del data set y se realizan las modificaciones necesarias, como la carga en un DataFrame y la manipulación de los datos.
2. Redes neuronales: En esta sección se divide el data set en conjuntos de entrenamiento y prueba, y se entrena y realiza predicciones utilizando el modelo MPLClassifier. Para evaluar el desempeño del modelo, se utilizan métricas como el accuracy score, f1 score, sensibilidad, especificidad y la curva ROC.
3. Máquina de soporte vectorial: En esta sección se sigue el mismo procedimiento que en la sección anterior, pero utilizando el modelo SVC (Máquina de Soporte Vectorial).
4. Naive Bayes: En esta sección se aplica el mismo procedimiento que en los modelos anteriores, pero utilizando el modelo GaussianNB (Naive Bayes).
5. Random Forest: En esta sección se aplica el mismo procedimiento que en los modelos anteriores, pero utilizando el modelo RandomForestClassifier (Random Forest).

II. Solución

Análisis exploratorio de los datos

Para este proyecto se tomó una base de datos con un total de 8124 muestras y 23 atributos, los cuales son:

Nombre de la columna	Descripción	Valores posibles
Class	Determina si el hongo es comestible o venenoso	venenoso, comestible
Cap-shape	Forma del píleo (sombrero) del hongo	campana, cónico, convexo, plano, nudoso, hundido
Cap-surface	Textura del píleo	vibroso, surcado, escamoso, suave
Cap-color	Color del píleo	café, canela, gris, ante, verde, rosado, morado, rojo, blanco, amarillo
Bruises	¿El hongo posee manchas?	sí, no
Odor	Tipo de olor del hongo	almendra, anís, creosota, de pescado, acre, rancio, sin olor, mordaz, picante
Gill-attachment	Morfología de las láminas del hongo	adjunto, decurrente, libre, onduladas
Gill-spacing	Distribución de las láminas	cercano, pegado, distante

Gill-size	Tamaño de las láminas	ancho, estrecho
Gill-color	Color de las láminas	negro,café, ante, chocolate, gris, verde, naranja, rosado, morado, rojo, blanco, amarillo
Stalk-shape	Forma del estípite del hongo	ancho, estrecho
Stalk-root	Raíz del estípite	bulboso,palo, taza, igual, rizomorfo, arraigada, sin raíz
Stalk-surface-above-ring	Textura del estípite encima del anillo	fibroso, surcado, escamoso, suave
Stalk-surface-below-ring	Textura del estípite debajo del anillo	fibroso, surcado, escamoso, suave
Stalk-color-above-ring	Color del estípite encima del anillo	café, canela, gris, ante, verde, rosado, morado, rojo, blanco, amarillo
Stalk-color-below-ring	Color del estípite debajo del anillo	café, canela, gris, ante, verde, rosado, morado, rojo, blanco, amarillo
Veil-type	Tipo del velo del hongo	parcial, universal
Veil-color	Color del velo	café, naranja, blanco, amarillo
Ring-number	Número de anillos	cero, uno, dos
Ring-type	Tipo de anillo	telaraña, evanescente, enardecido, largo, colgante, revestido, separado
Spore-print-color	Color de la esporada que libera el hongo	negro, café, ante, chocolate, verde, naranja, morado, blanco, amarillo
Population	Distribución de la población del hongo	abundante, agrupada, numerosa, repartida, solitaria, poca
Habitat	Hábitat del hongo	planta, hoja, prado, tierra, ciudad, basura, bosque

Tabla 1. Descripción de la base de datos.

Limpieza de datos

Primero se verificó si el dataset contiene valores nulos o atípicos. Para ello, se utilizó el método DataFrame.info() de la biblioteca pandas.

En el caso de este dataset, no se encontraron categorías con valores nulos o atípicos. De esta manera, no hubo necesidad de tratar valores faltantes.

Lo segundo fue la recolección de los datos únicos de cada columnas, a través del método Numpy.unique().

```
1 #Información general del Dataset
2 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8124 entries, 0 to 8123
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Class                     8124 non-null   object
 1   Cap-shape                 8124 non-null   object
 2   Cap-surface               8124 non-null   object
 3   Cap-color                 8124 non-null   object
 4   Bruises                   8124 non-null   object
 5   Odor                      8124 non-null   object
 6   Gill-attachment           8124 non-null   object
 7   Gill-spacing              8124 non-null   object
 8   Gill-size                 8124 non-null   object
 9   Gill-color                8124 non-null   object
 10  Stalk-shape               8124 non-null   object
 11  Stalk-root                8124 non-null   object
 12  Stalk-surface-above-ring  8124 non-null   object
 13  Stalk-surface-below-ring  8124 non-null   object
 14  Stalk-color-above-ring    8124 non-null   object
 15  Stalk-color-below-ring    8124 non-null   object
 16  Veil-type                 8124 non-null   object
 17  Veil-color                8124 non-null   object
 18  Ring-number               8124 non-null   object
 19  Ring-type                 8124 non-null   object
 20  Spore-print-color         8124 non-null   object
 21  Population                8124 non-null   object
 22  Habitat                   8124 non-null   object
dtypes: object(23)
memory usage: 1.4+ MB
```

```
1 for i in ds:
2   print(i,":",ds[i].unique().size)
```

```
Class : 2
Cap-shape : 6
Cap-surface : 4
Cap-color : 10
Bruises : 2
Odor : 9
Gill-attachment : 2
Gill-spacing : 2
Gill-size : 2
Gill-color : 12
Stalk-shape : 2
Stalk-root : 5
Stalk-surface-above-ring : 4
Stalk-surface-below-ring : 4
Stalk-color-above-ring : 9
Stalk-color-below-ring : 9
Veil-type : 1
Veil-color : 4
Ring-number : 3
Ring-type : 5
Spore-print-color : 9
Population : 6
Habitat : 7
```

En este paso se aprecia que la categoría ‘Veil-type’ solo tiene un valor, por lo cual se decide eliminar esa columna del data set, ya que no es significativa para el análisis.

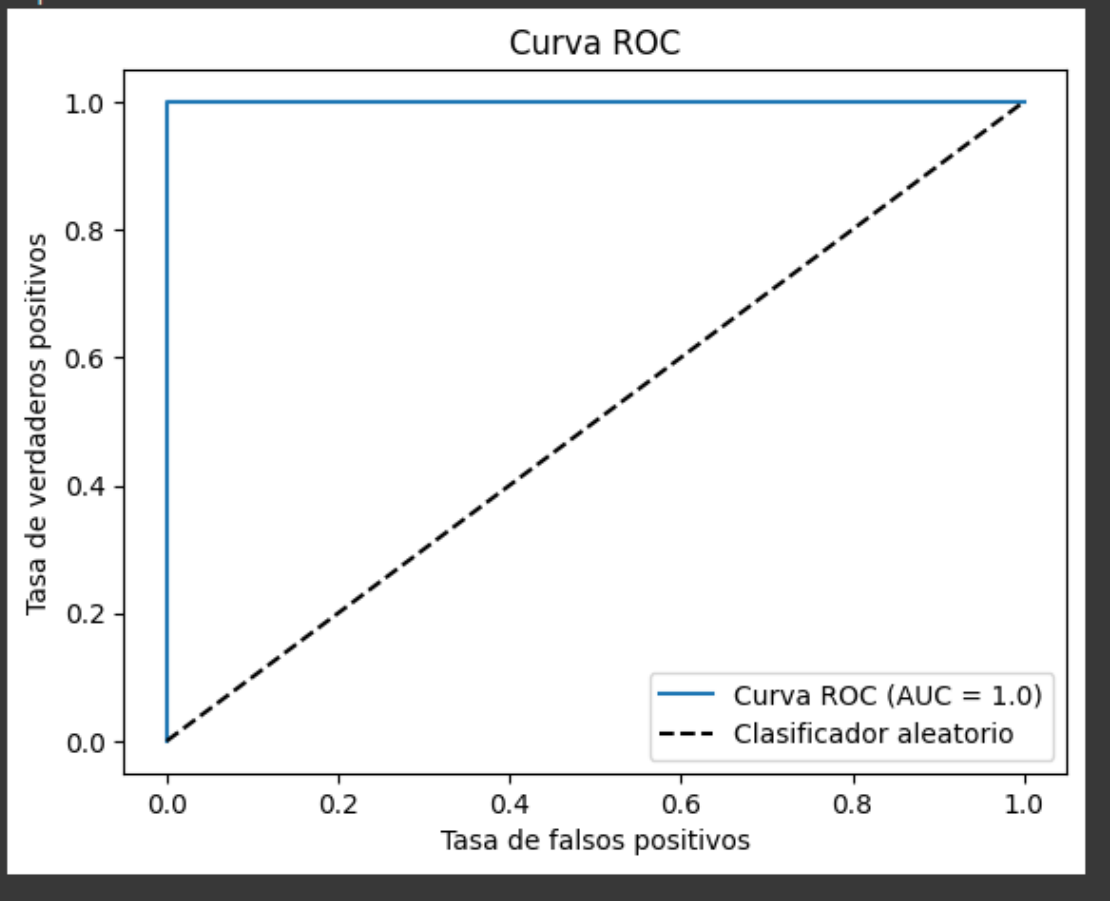
```
1 #Se elimina la columna Veil-type
2 df = df.drop("Veil-type", axis=1)
```

III. Iteraciones

Redes Neuronales

El modelo de Redes Neuronales obtuvo una precisión del 100%, lo cual indica que clasificó correctamente todos los ejemplos en el conjunto de prueba. Además, el puntaje F1 de 1.0, la sensibilidad de 1.0 y la especificidad de 1.0 sugieren que el modelo fue capaz de predecir tanto las muestras positivas como las negativas de manera perfecta. Estos resultados indican un rendimiento sobresaliente del modelo y sugieren que las Redes Neuronales son altamente efectivas para este conjunto de datos.

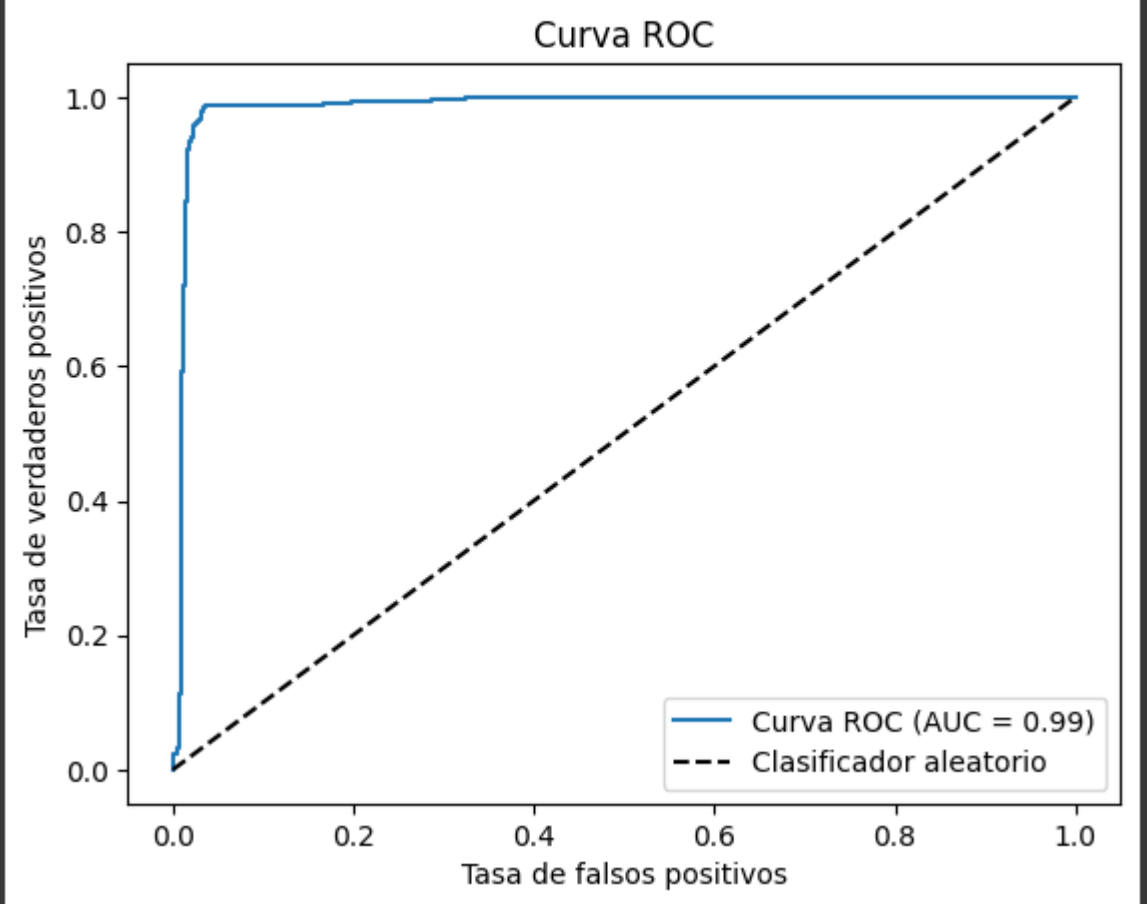
```
**ENTRENAMIENTO CON REDES NEURONALES**
Precisión del modelo: 100.0%
Puntaje F1 del modelo: 1.0
Sensibilidad: 1.0
Especificidad: 1.0
```



Máquina de Soporte Vectorial

El modelo de Máquina de Soporte Vectorial logró una precisión del 97.29%, lo que indica que clasificó la mayoría de los ejemplos de manera correcta. El puntaje F1 de 0.972 y la sensibilidad de 0.99 indican un buen equilibrio entre la precisión y la exhaustividad del modelo. La especificidad de 0.96 sugiere que el modelo pudo identificar correctamente la mayoría de las muestras negativas. En general, estos resultados muestran un rendimiento sólido de la Máquina de Soporte Vectorial en este conjunto de datos.

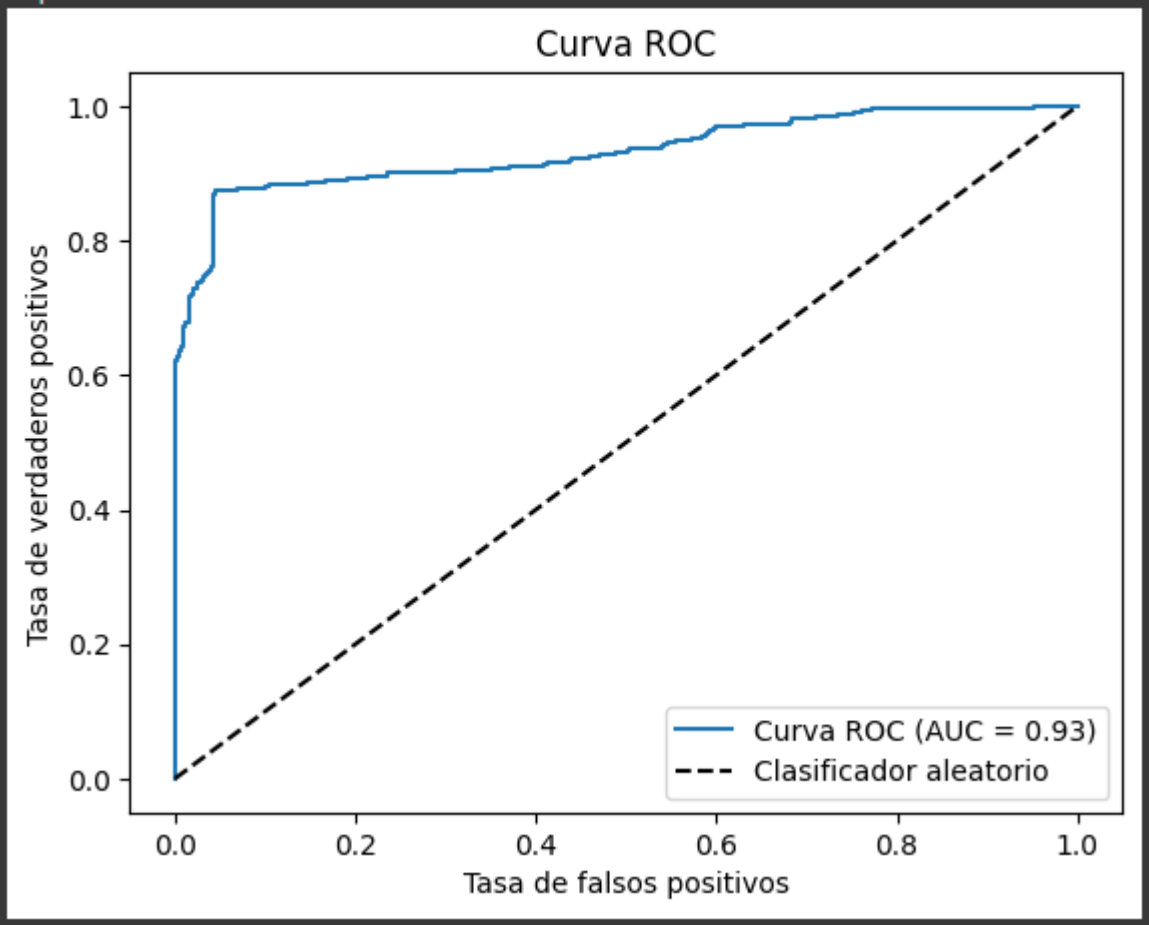
```
**ENTRENAMIENTO CON MÁQUINA DE SOPORTE VECTORIAL**
Precisión del modelo: 97.29%
Puntaje F1 del modelo: 0.9724310776942355
Sensibilidad: 0.99
Especificidad: 0.96
```



Naive Bayes

El modelo de Naive Bayes obtuvo una precisión del 88.74%, lo que indica que clasificó correctamente la mayoría de los ejemplos en el conjunto de prueba. El puntaje F1 de 0.883, la sensibilidad de 0.88 y la especificidad de 0.89 sugieren un equilibrio aceptable entre precisión y exhaustividad. Aunque estos resultados son inferiores a los obtenidos por los otros dos clasificadores anteriores, aún indican un rendimiento decente de Naive Bayes en este conjunto de datos.

```
**ENTRENAMIENTO CON NAIVE BAYES**
Precisión del modelo: 88.74%
Puntaje F1 del modelo Naive Bayes: 0.8829174664107485
Sensibilidad: 0.88
Especificidad: 0.89
```



IV. Resultados

En general, podemos concluir que tanto las Redes Neuronales como la Máquina de Soporte Vectorial son modelos eficaces para este conjunto de datos, con rendimientos destacados en términos de precisión, puntaje F1 y capacidad para predecir correctamente las muestras positivas y negativas. Por otro lado, Naive Bayes también muestra un rendimiento aceptable, aunque ligeramente inferior a los otros dos clasificadores.

V. Enlaces a contenido externo

Dataset: Mushroom Classification | Kaggle

Video: https://youtu.be/jPth-E2NGwE