Hongos? Comestibles o Venenosos?

- **Profesor: Phd. JORGE ALEXIS CASTILLO SEPULVEDA**
- Alumnno: Didimo Escobar
- Fecha: 09 de septiembre de 2023

ABSTRACT

En este análisis, se ejecuta un modelo de clasificación con datos que intentan clasificar los hongos como venenosos o comestibles. Los datos en sí son en su totalidad nominales y categóricos. Los datos provienen un repositorio de aprendizaje automático de la UCI Input Profesor Jorge Castillo (https://archive.ics.uci.edu/dataset/848/secondary+mushroom+dataset). Los objetivos consistian en 'desplegar' un clasificador que permita decidir si un hongo es comestible o venenoso a partir de sus atributos.

Contexto General de diseño

Introducción

La navaja de Occam, también conocida como ley de la parsimonia, es quizás uno de los principios más importantes de toda la ciencia. La teoría basada en la menor cantidad de supuestos tiende a ser la correcta. En el caso del aprendizaje automático, se podría proponer una condición corolaria; *los mejores modelos de aprendizaje automático no solo requieren las mejores métricas de rendimiento, sino que también deben requerir la menor cantidad de datos y tiempo de procesamiento.*

En este análisis, mi `espiritu` fue construir un modelo con las métricas de rendimiento más altas (precisión y F1 Score) utilizando la menor cantidad de datos y operando en el menor tiempo. Quizás menciono el menor tiempo ya que, es hora de que los modelos sean procesados en notebooks con Databricks, Snowflake, en donde es posible pasar mayor recurso de procesamiento y sin dejar de lado el procesar en pyspark que es mas funcional.

En conjunto, quería determinar cuáles eran los factores clave al clasificar un hongo como venenoso o comestible.

Información General

Categorizar algo como venenoso versus comestible no sería un problema tomado a la ligera. Si tuviera algún margen de error, alguien podría morir. Por lo tanto, cualquier modelo que prediga si un hongo es venenoso o comestible debe tener una precisión perfecta. A primera vista, este es el objetivo de los datos: determinar qué comer y qué botar literalmente. Un problema típico en la clasificación.

Data pre-processing

Obviamente, un modelo de aprendizaje automático no sería capaz de procesar letras cuando debería haber números, por lo que se justificaba un proceso de codificación. Usando la función pandas `.get_dummies()` pude generar una tabla llena de datos completamente binarios, donde 1 está presente si una característica de una columna determinada estaba presente y 0 en caso contrario.

Después de convertir al formato binario, las 21 columnas originales se transformaron en 116 columnas. No se eliminaron filas.

Identificando features irrelevantes o menos importantes

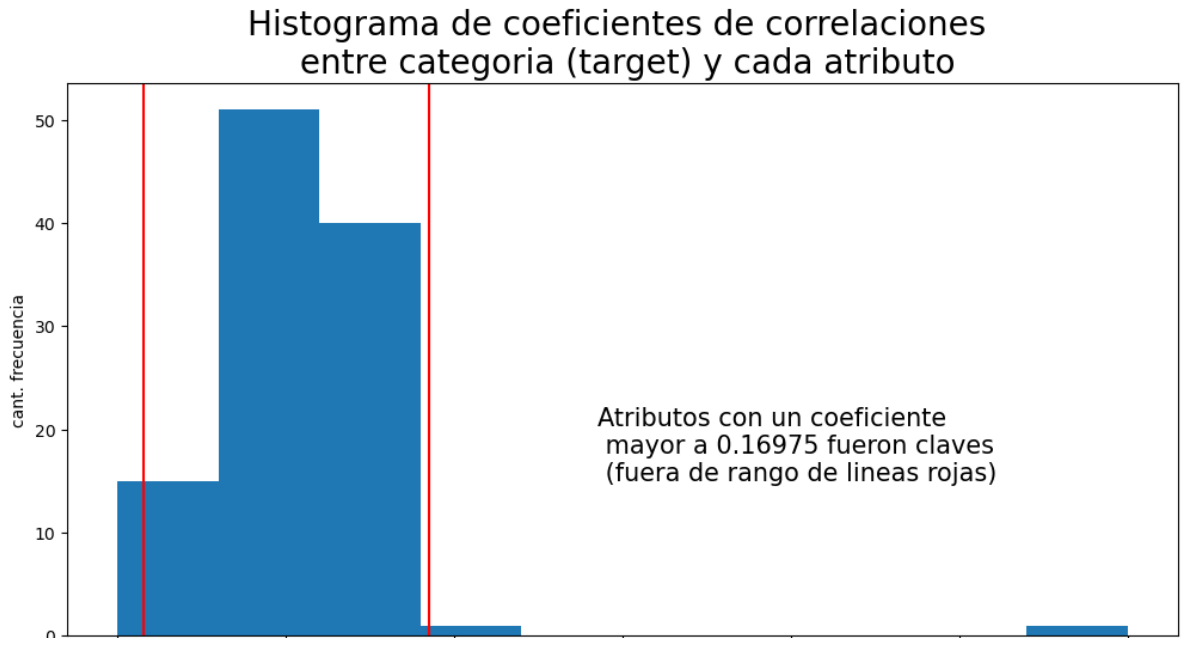
Las features selection se tomaron en función de los métodos de filtrado. Pero antes de determinar el nivel de influencia de cada feature, quería saber qué características eran totalmente inútiles. Para ello se utilizaron dos métodos.

1. Prueba de hipótesis de Chi-Cuadrado, sobre los datos en su forma original (se encontraron 4 características irrelevantes).
2. Prueba de hipótesis Bootstrap de la diferencia media de cada característica entre los venenosos y los comestibles, después de que los datos se convirtieron a forma binaria (se encontraron 8 características irrelevantes).

En ambos casos, la hipótesis nula fue que la distribución de una característica NO era la misma tanto para los hongos comestibles como para los venenosos. Un bucle for actuó en todas las funciones en el formato limpio y se realizaron pruebas de hipótesis en cada una. Después de que se encontraron características inútiles, se descartaron (las elimine). `En total, se encontró que las doce (12) características eran irrelevantes y no tenían influencia para determinar la categoría` . Luego, los datos para el modelado se redujeron a 108 columnas.

Selección de features importantes con filtrado

Una vez que los datos estuvieron en forma binaria, se realizó un gráfico de histograma entre la correlación de cada característica y la clase (el target). Utilizando los valores de las correlaciones, se realizó un proceso de prueba y error ajustando una variedad de modelos de clasificación a un conjunto de características que tenían una magnitud (valor absoluto) mayor que un valor umbral de correlación. Se descubrió que todo el conjunto de características con una magnitud mayor que $|\pm 0.169757|$ eran datos suficientes para producir un modelo que funcionó con perfecta precisión en una división de prueba de 70-30.



Feature Ranking

After converting into binary form, features were then fed into the models and ranked descendingly in accordance to the magnitude of their correlation coefficient with the target variable, `class` . Thus the first feature fed into the model had the highest magnitude of correlation, the second had the second highest, and so on. The first five rows of the feature rank table looked like this;

Después de convertirlas a forma binaria, las características se introdujeron en los modelos y se clasificaron de manera descendente de acuerdo con la magnitud de su coeficiente de correlación con la variable objetivo, `class` . Así, la primera característica incorporada al modelo tenía la mayor magnitud de correlación, la segunda tenía la segunda más alta, y así sucesivamente. Las primeras cinco filas de la tabla de clasificación de características tenían este aspecto:

Rank	Feature
1	stem-color_w
2	stem-surface_s
3	gill-attachment_p
4	ring-type_z
5	stem-surface_g

Selección de Modelo

Se eligieron múltiples modelos para la evaluación. Estos incluyeron:

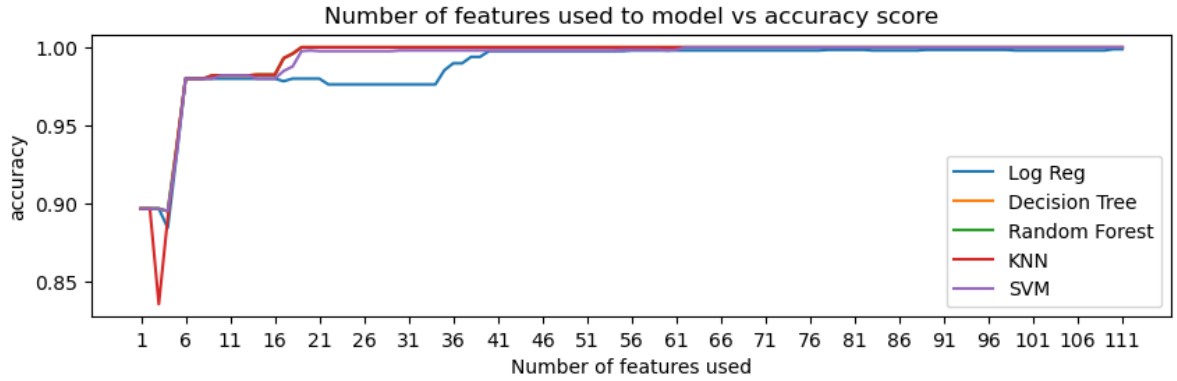
- logistic regression
- K-nearest-neighbors
- support vector machine
- decision tree classifier
- random forest classifier

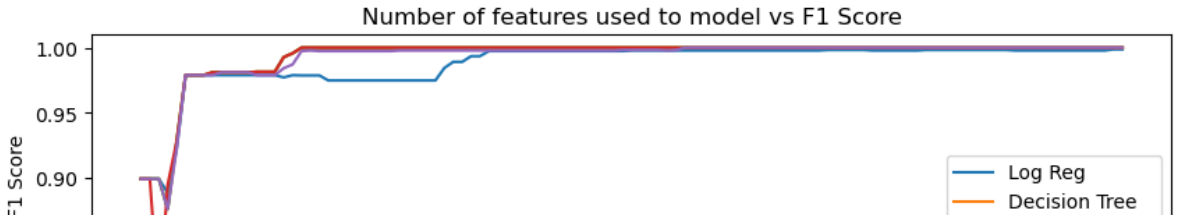
El rendimiento del modelo se evaluó en:

- Accuracy score
- F1 score
- Tiempo combinado de entrenamiento más predicción.

Cada modelo fue alimentado a través del bucle for mencionado anteriormente y evaluado en una prueba dividida de 70-30. Como podemos ver en los gráficos a continuación, fueron las 19 características mejor clasificadas las que la mayoría de los modelos comenzaron a calificar con perfecta precisión.

A pesar de que el bosque aleatorio, los k-vecinos más cercanos y los árboles de decisión obtuvieron puntuaciones perfectas cuando se alimentaron con 19 características, fueron los árboles de decisión los que se desempeñaron en el menor tiempo. *Por lo tanto, el clasificador de árbol de decisión fue el mejor modelo*.





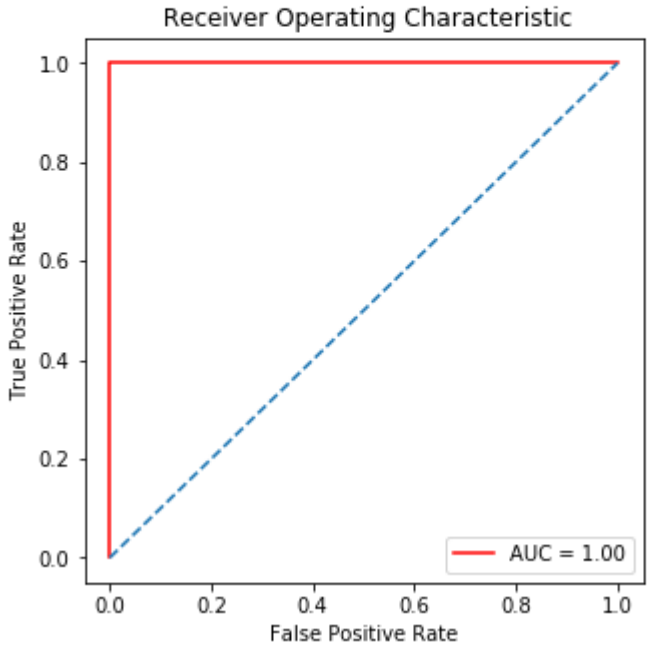
Evaluación de Modelo: Decision Tree Classifier

El clasificador de árbol de decisión fue el modelo que cumplió con los criterios de desempeño en la menor cantidad de tiempo, con la menor cantidad de características y con métricas de desempeño máximas en F1 y puntajes de precisión.

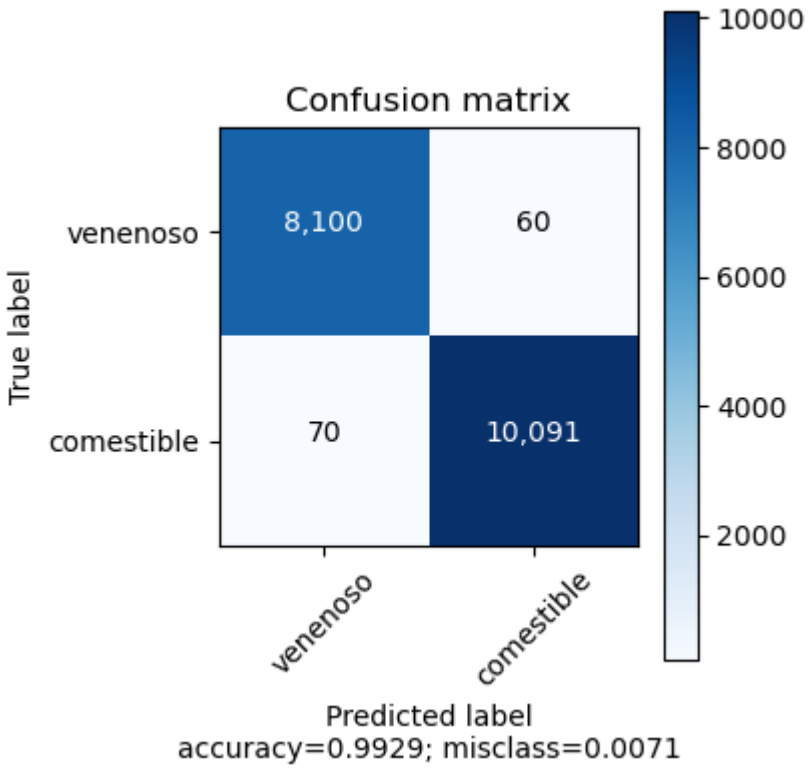
Metric	Value
Accuracy	0.992904
F1 score	0.9936
AUC	1.0000
Numero de features usados	90
tiempo train+predict	0.234376 (s)
Test set size	(30%)
Train set size	(70%)

Especificamente, los hiperparametros y roc-auc fueron:

```
DecisionTreeClassifier(class_weight='balanced', criterion='gini',
    max_depth=None, max_features=None, max_leaf_nodes=None,
    min_impurity_decrease=0.0, min_impurity_split=None,
    min_samples_leaf=1, min_samples_split=2,
    min_weight_fraction_leaf=0.0, presort=False, random_state=None,
    splitter='best'):
```



With a confusion matrix of;



Comentarios sobre accuracy metrics

Aunque no es común obtener puntuaciones perfectas en los modelos, sucede. Acá entra un debate frente a este modelo, en donde claramente un equipo multidisciplinario podrá entregar

Conclusión y reflexiones

Usando solo 17 features, es posible :) concluir con 100% de certeza que un hongo es comestible o venenoso. Estas son los 20 features, clasificados en orden descendente según el valor absoluto con su correlación con la clase objetivo. Como input la clase objetivo o target, comestible se marcó con 0 y venenoso con 1.

Una correlación negativa significa que si un hongo tiene esa característica es más probable que sea comestible

Una correlación positiva significa que si un hongo tiene esa característica, es más probable que sea venenoso.

Rank	Feature
1	stem-color_w
2	stem-surface_s
3	gill-attachment_p
4	ring-type_z
5	stem-surface_g
6	cap-surface_k
7	cap-shape_b
8	stem-root_r
9	gill-color_w
10	cap-color_e
11	gill-attachment_a
12	stem-root_f
13	stem-surface_f
14	stem-color_f
15	spore-print-color_n
16	gill-color_n
17	cap-color_r
18	gill-attachment_e
19	gill-spacing_d
20	cap-color_n

Interpretación de estas características

Decision tree model tiene un flujo de trabajo que nos ayuda a sacar conclusiones.

Comenzando en la parte superior, para una fila determinada (es decir, un hongo determinado) si la característica `odor_n <=0.5` (que en realidad significa `odor_n = 0` o `odor_none=False`, o *tiene un olor*) Y `bruises_t <=0.5` (es decir, `bruises_t = 0` o, el *hongo NO causa moretones*), entonces concluimos que el hongo es venenoso. Se pueden sacar más conclusiones simplemente siguiendo el árbol.