ORANGE:

Empezamos cargando y preprocesando el csv, de la siguiente manera:

Seleccionamos la variable target, en este caso uso el tipo dado que sino más adelante da error, tras esto normalizo y discreto el dataset:

Una vez hecho esto, puedo empezar a sacar info del dataset, por ejemplo:

Creo una Data table a partir de los datos:

Vemos más gráficas:

Etc.

Tras esto creamos el árbol de decisión:

Lo visualizamos:

Y vemos la informacción del modelo:

A partir de aquí, no se puede hacer mucho más, no están los widgets necesarios que se pedían en la práctica, el esquema en Orange quedaría tal que así:

Dado que esto no me ha llevado muy lejos, procedo a intentarlo con un script de python:

import Orange

# Paso 1: Cargar los datos en Orange

data = Orange.data.Table("/Users/javipozo/Desktop/ProyectosPython/tomates/Orange/datos.csv")

#Paso 2: Discretizar variables y seleccionar la variable objetivo

data\_new = Orange.data.Table("/Users/javipozo/Desktop/ProyectosPython/tomates/Orange/datos\_orange.csv")

discretizer = Orange.preprocess.Discretize()

data = discretizer(data\_new)

rend\_var = data.domain["rendimiento"]

data.domain.class\_var = rend\_var

# Imprimir información sobre la variable de clase discretizada

print("Variable de clase discretizada:", data.domain.class\_var.name)

print("Valores posibles:", data.domain.class\_var.values)

# Paso 3: Construir el modelo de árbol de decisión

model = Orange.classification.tree.TreeLearner()(data)

print(model)

# Realizar una exploración inicial de los datos

print(f"Number of instances: {len(data)}")

print(f"Number of attributes: {len(data.domain.attributes)}")

print(f"Attribute names: {data.domain.attributes}")

import random

n = len(data)

proporcion\_entrenamiento = 0.8 # Por ejemplo, 80% para entrenamiento

# Paso 4: Obtener índices aleatorios para la muestra de entrenamiento

indices\_entrenamiento = random.sample(range(n), int(n \* proporcion\_entrenamiento))

# Paso 5: Crear conjuntos de entrenamiento y prueba

train\_data = Orange.data.Table(data.domain, [data[i] for i in indices\_entrenamiento])

test\_data = Orange.data.Table(data.domain, [data[i] for i in range(n) if i not in indices\_entrenamiento])

# Paso 6: Imprimir el tamaño de los conjuntos de entrenamiento y prueba

print("Tamaño del conjunto de entrenamiento:", len(train\_data))

print("Tamaño del conjunto de prueba:", len(test\_data))

model = Orange.classification.tree.TreeLearner()(test\_data)

print(model)

#De aqui en adelante orange no funciona, no tiene los modelos y saltan errores en las prediciones por modulos que no tiene ya sea por el sistema operativo de mi ordenador, la version instalada, o causas externas.

#predictions = model(test\_data)

#accuracy = Orange.evaluation.Accuracy(predictions, test\_data)

#print(f"Accuracy: {accuracy}")

En este código, empezamos leyendo los datos y una vez leídos, procedo a discretizar variables y seleccionar la variable objetivo, una vez eso funciona, creo el árbol de decisión, tras esto, separo los datos en entrenamiento y prueba, y vuelvo a hacer el árbol con estos datos; al intentar avanzar de aquí empiezan a salir errores muy raros que no he sabido gestionar, como:

ValueError: max() arg is an empty sequence

Y estos errores no me dejaban hacer las prediciones correspondientes, a parte siguiendo con la práctica, saltan otros errores de atributos como:

AttributeError: module 'Orange.classification' has no attribute 'CARTLearner'

Y no se puede seguir dado que no tiene los modulos nececesarios, asique procedo a terminar lo que pide la práctica en código python que esta en el archivo an.ipynb.