id3aa

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier,plot_tree
import matplotlib.pyplot as plt
import math

# Load the dataset
df = pd.read_csv('diabetes.csv')

# Display the first few rows of the dataset
print(df.head())

def calculate_entropy(data, target_column):
    total_rows = len(data)
    target_values = data[target_column].unique()

    entropy = 0
    for value in target_values:
        # Calculate the proportion of instances with thecurrent value
        value_count = len(data[data[target_column] ==value])
        proportion = value_count / total_rows
        entropy -= proportion * math.log2(proportion) ifproportion != 0
else 0

    return entropy

entropy_outcome = calculate_entropy(df, 'Outcome')
print(f"Entropy of the dataset: {entropy_outcome}")

def calculate_information_gain(data, feature,target_column):
    # Calculate weighted average entropy for thefeature
    unique_values = data[feature].unique()
    weighted_entropy = 0

    for value in unique_values:
        subset = data[data[feature] == value]
        proportion = len(subset) / len(data)
        weighted_entropy += proportion *calculate_entropy(subset,
target_column)

    # Calculate information gain
    information_gain = entropy_outcome –weighted_entropy

    return information_gain

# Calculate entropy and information gain for each feature
for column in df.columns[:-1]:
    entropy = calculate_entropy(df, column)
    information_gain = calculate_information_gain(df, column,
'Outcome')
    print(f"{column} – Entropy: {entropy:.3f}, Information Gain:
{information_gain:.3f}")
```