Implementation of ID3 Decision Tree Algorithm

https://colab.research.google.com/drive/1jC9zWP-wBXuYrG6lD92ha3SaP6Tz1eoY?usp=sharing

Introduction:

A decision tree is a tree where each node represents a feature (attribute), each link (branch) represents a decision rule, and each leaf represents an outcome for a given scenario. The data used is a classification dataset, it is about a mass collected from the breast of an individual and measurements collected, such as mean radius, mean perimeter, mean smoothness etc. Help evaluate whether or not that mass is malignant of benign.

The report presents an implementation of a decision tree alongside entropy, to help predict the classification problem where the mass is either malignant or benign. The goal of the decision tree is to create a training set that can be used to predict the value of the target by learning simple decision rules.

ID3 Algorithm:

The ID3 algorithm builds decision trees using a top-down approach through multiple branches with no backtracking which may sometimes be considered a 'greedy approach' since there is no back tracking allowed.

The ID3 Algorithm consists of the following modules: Entropy computation and information gain computation

Entropy:

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

Entropy refers to a measure of randomness present in the data being handled. The higher the entropy, the more difficult it is to draw any conclusions from the dataset and may make it unreliable. Id3 follows a rule where a branch with an entropy of zero is a leaf node and a branch with entropy more than zero needs to be further split. Helps decide how to grow the tree or how to split it.

Optimizing Decision Tree Using Entropy

```
[ ]  from sklearn.tree import DecisionTreeClassifier
     clf = DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=1)
     clf = clf.fit(X_train, y_train)
```

The 'criterion' by default is 'gini', this parameter allows us to use different attribute selection measure and 'entropy' is used for the information gain.

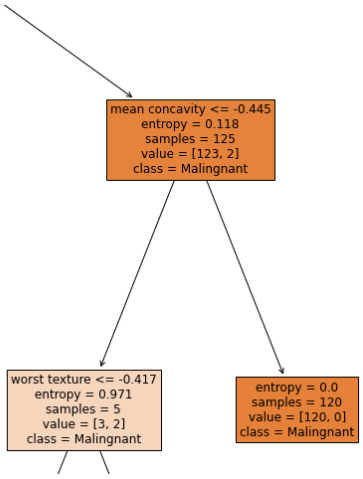
Information Gain:

Information gain is a statistical property that measures how well attributes separate the training set according to their target classifications. A higher amount of information gain with have a lower

entropy, it calculates the difference between entropy before the split in the tree and average entropy after split of the dataset based on values provided.

$$Information\ Gain = Entropy(before) - \sum_{j=1}^{K} Entropy(j, after)$$

Where 'before' is the dataset before the split in the dataset, k is the number of subsets generated because of the split and 'j after' is the subset after the split in data.



The image to the left displays a branch of the decision tree. As we can see, lower the entropy level, higher the information gained for the user. A branch is concluded once the entropy level of 0 has been reached since we can clearly understand the information, we receive from that. In the example on the left, the branch on the further right reaches an entropy level of 0 which tells us that the mass collected from an individual is most likely malignant.

Model Evaluation:

```
data = load_breast_cancer()
dataset = pd.DataFrame(data=data['data'], columns=data['feature_names'])
dataset
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness
0	17.99	10.38	122.80	1001.0	0.11840	0.27760
1	20.57	17.77	132.90	1326.0	0.08474	0.07864
2	19.69	21.25	130.00	1203.0	0.10960	0.15990
3	11.42	20.38	77.58	386.1	0.14250	0.28390
4	20.29	14.34	135.10	1297.0	0.10030	0.13280
...	...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	0.11100	0.11590
565	20.13	28.25	131.20	1261.0	0.09780	0.10340
566	16.60	28.08	108.30	858.1	0.08455	0.10230
567	20.60	29.33	140.10	1265.0	0.11780	0.27700
568	7.76	24.54	47.92	181.0	0.05263	0.04362

569 rows × 30 columns

The model is evaluated on the breast cancer dataset. It contains 569 rows of data with 30 columns (variables). These variables help determine whether an individual will have a malignant of benign mass. The decision tree will help determine which factors or variables help predict the results the best.

Data Preparation:

The dataset includes attributes that play into factor when trying to assess whether a breast mass is either malignant or benign.

The function ‘dataset.shape’ tells us that there are 569 rows with 30 attributes. Some attributes used are mean radius of mass, mean texture, mean perimeter etc. And the command ‘dataset.describe’ is used to gives us a brief summary statistic of the dataset and all the attributes present in the dataset.

The last step of data preparation is checking the data set for missing values, the function used is ‘dataset.isnull().sum(). As we can see, there are no missing values in the dataset.

```
dataset.shape   #Rows and Attributes

(569, 30)
```

```
[38] dataset.isnull().sum() # Check

mean radius                0
mean texture               0
mean perimeter             0
mean area                  0
mean smoothness            0
mean compactness           0
mean concavity             0
mean concave points        0
mean symmetry              0
mean fractal dimension     0
radius error               0
texture error              0
perimeter error            0
area error                 0
smoothness error           0
compactness error          0
concavity error            0
concave points error       0
symmetry error             0
fractal dimension error    0
worst radius               0
worst texture              0
worst perimeter            0
worst area                 0
worst smoothness           0
worst compactness          0
worst concavity            0
worst concave points       0
worst symmetry             0
worst fractal dimension    0
dtype: int64
```

Experiment Design and Evaluation:

In the first round of testing, the data is split into two parts, the training set and the test set which is used for the evaluation of the findings. ‘test_size=0.33’, set aside how much of the data we want to test to see if it’s performing well or not, in this case 33% of the original dataset has been used for testing.

Some parameters such as ‘max_depth’ have been used, ‘max_depth’ regulates how deep the tree can be, if there’s 1 node then it’s only a depth of 1, if it’s 2 decisions with 2 nodes, that’s a depth of 2. Not something you can know beforehand. Trying different values helps to see which one works best, for this data set, 4 has been used.

```
[3] from sklearn.model_selection import train_test_split
    X = dataset.copy()
    y = data['target']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)
```

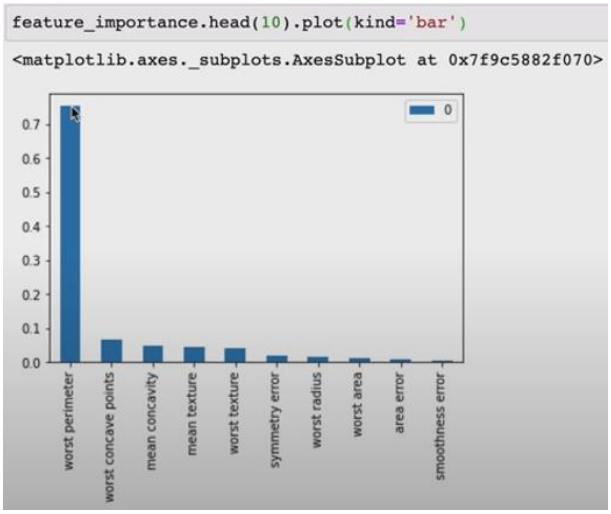
```
[4] X_test
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	c
271	11.290	13.04	72.23	388.0	0.09834	0.07608	
428	11.130	16.62	70.47	381.1	0.08151	0.03834	
46	8.196	16.84	51.71	201.9	0.08600	0.05943	
345	10.260	14.71	66.20	321.6	0.09882	0.09159	
112	14.260	19.65	97.83	629.9	0.07837	0.22330	
...	...	...	...	...	...	...	
515	11.340	18.61	72.76	391.2	0.10490	0.08499	
344	11.710	15.45	75.03	420.3	0.11500	0.07281	
482	13.470	14.06	87.32	546.3	0.10710	0.11550	
335	17.060	21.00	111.80	918.6	0.11190	0.10560	
286	11.940	20.76	77.87	441.0	0.08605	0.10110	

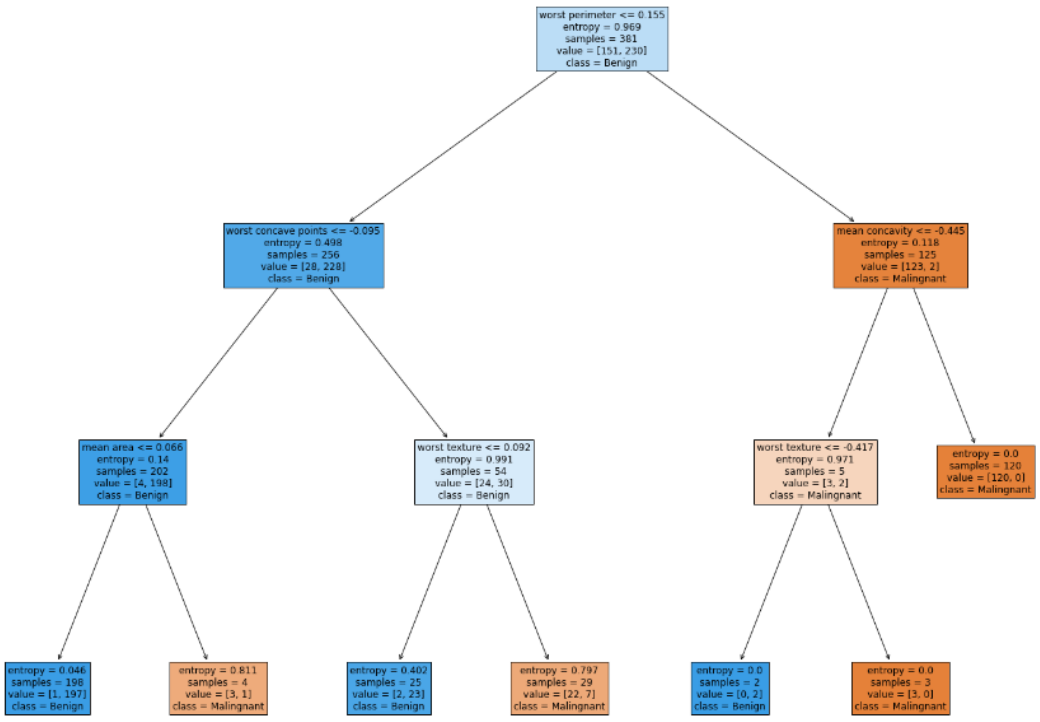
188 rows × 30 columns

Evaluation Results:

Using the test set, a bar chart has been made using all the parameters, and we can conclude that the attribute or variable called 'worst perimeter' is the best at determining if a tumor is malignant or benign.



Another way to see how the dataset or how the model is working is to create a plot of the decision tree. A tree is created on how the model decides on a result. When you get one data point, and it considers if the value is greater or lower than and decides on another decision point. As far as the depth goes, it is limited to 4 levels because of the parameters set by us. Pruning is an important way of making sure the decision tree is not overfeeding to our dataset, it makes our model less complex, more explainable, and easier to understand than an unpruned model.



Conclusion:

We can conclude that decision trees are a great in compared to other algorithms, in the sense that they require less effort for data preparation. A decision tree is also very intuitive and easy to explain on a technical level to stakeholders of a business or for the purpose of research. However, decision trees can be inadequate to predict continuous values and a small change in the data could shift the structure of the tree in its entirety.

In our case, to predict the probability of an individual having a malignant or benign tumor, a decision tree algorithm is best fit to resolve this. A decision tree also helps determine which factors have the biggest effect on someone since those attributes would have the shortest branch. Commands can be run to check the accuracy of each of these results.