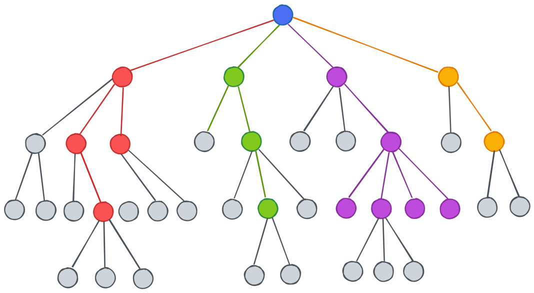


A decision tree consists of one root node, decision nodes and leaf nodes.

- **Root node:** We begin with asking "which attribute should be tested at the root?" This attribute will become our root node. Here, the entire population is being analysed.
- **Decision nodes:** Here, we make 'splits' in our data and check conditions to further divide our dataset for classification. They are based on attributes.
- **Leaf node or terminal node:** They are nodes at the end of the tree which do not split into more nodes. they each consist of a class.

Measuring Impurity in a node

A node's purity is measured on the basis of how many datapoints of different categories are present in its region.
Let $p_{i,j}$ be the proportion of data points in node i belonging to class j. Some of the common meaures of impurity are:

- **Gini Index:** We classify observations to class j with probability $p_{i,j}$. Formula of Gini Index is given by

$$G_i = \sum_{j=1}^{J} p_{i,j}(1 - p_{i.j})$$

- **Entropy:**It is the disorder among a node. It is given by the formula:

$$H_i = -\sum_{j=1}^{n} p_{i,j} \; \log_2 p_{i,j}$$

- **Information Gain:** It is the expected reduction in entropy.

$$\text{IG(attr) = entropy of dataset - entropy of attribute}$$

Training

Step 1: Select the root node, or the best attribute for splitting the data. This is done using feature selection which utilizes impurity measures.
Step 2: Divide the root node into subsets that contains possible values for the best attributes.
Step 3: Generate the decision tree node, which contains the best attribute.
Step 4: Repeat from step two but using the current most leaf nodes until we reach a stopping criterion.

When do we stop splitting?

There are a few stopping points, and we stop if we reach any one of them.

- Every observation in the node belong to the same class.
- There are no more attributes on the basis of which we can make a split. Here, we label all the observations with the majority class.
- We are left with no more observations in the node.

One of the biggest disadvantages of Decision trees is that they have high variance due to being non parametric. To rectify this, we use ensembles.

Random Forest

Ensembles use multiple models and combine their predictions to get better performace as compared to individual models.
Random forest is an Ensemble learning method that uses multiple decision trees to reduce variance and get better performace. A random forest algorithm uses bagging with decision trees.

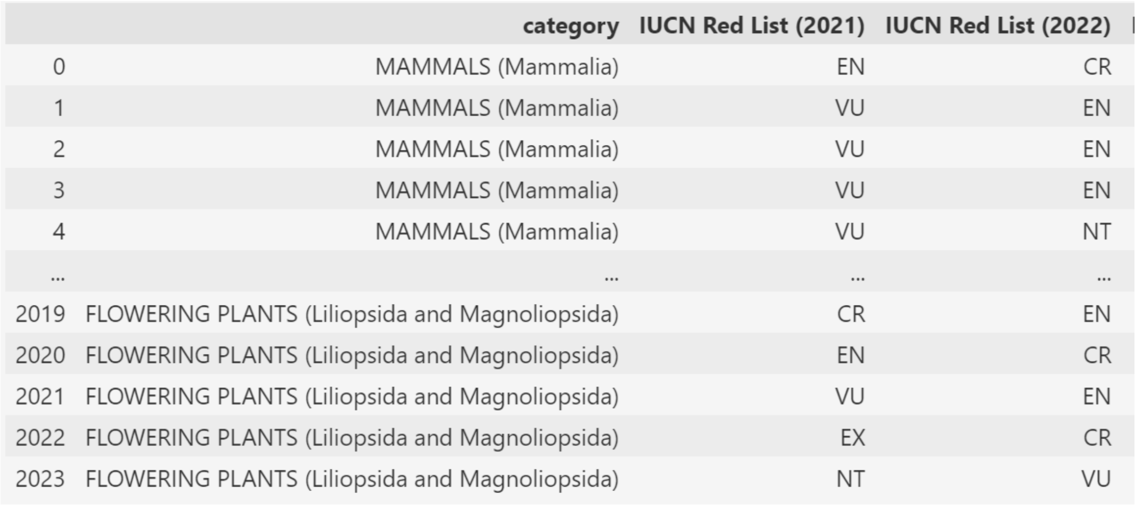
Step 1: Take k samples from our dataset, $D_1, D_2, ..., D_k$ with replacement.
Step 2: On each D_i, we train a Decision tree, but only taking some of the features for splitting in each one. **Step 3:** Our model is:

$$model = \frac{1}{k}\sum_{j=1}^{k} h_j(x)$$

Random Forest Using Sci-kitlearn

Sklearn implements the CART algorithm by default for training decision trees as well as random forest.

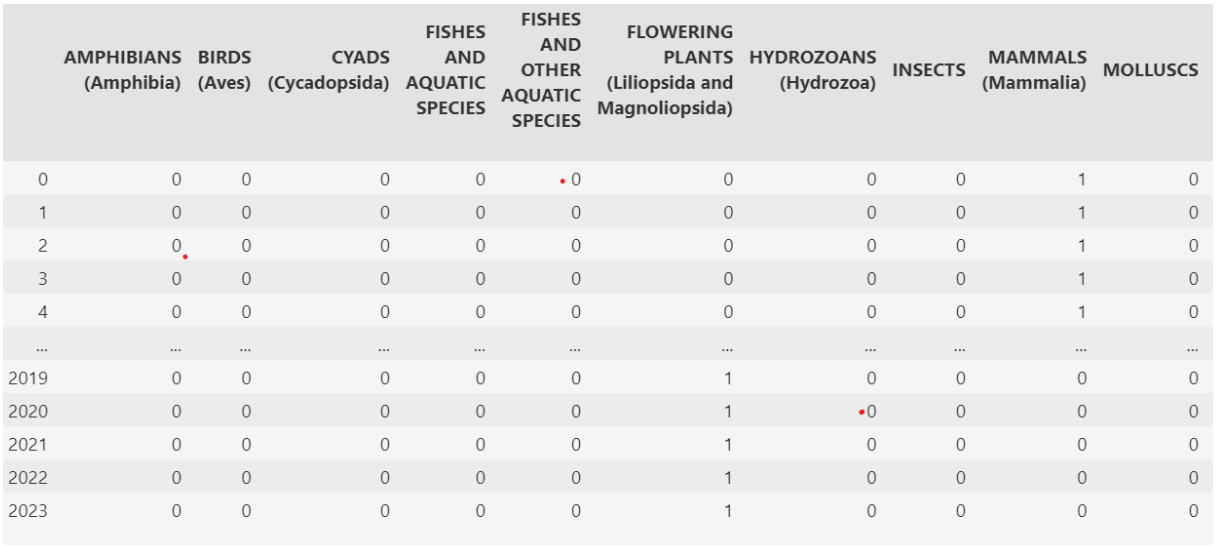
e Our dataset:



	category	IUCN Red List (2021)	IUCN Red List (2022)
0	MAMMALS (Mammalia)	EN	CR
1	MAMMALS (Mammalia)	VU	EN
2	MAMMALS (Mammalia)	VU	EN
3	MAMMALS (Mammalia)	VU	EN
4	MAMMALS (Mammalia)	VU	NT
...	...	...	...
2019	FLOWERING PLANTS (Liliopsida and Magnoliopsida)	CR	EN
2020	FLOWERING PLANTS (Liliopsida and Magnoliopsida)	EN	CR
2021	FLOWERING PLANTS (Liliopsida and Magnoliopsida)	VU	EN
2022	FLOWERING PLANTS (Liliopsida and Magnoliopsida)	EX	CR
2023	FLOWERING PLANTS (Liliopsida and Magnoliopsida)	NT	VU

We can see that both our features and labels are categorical variables. sklearn used an optimized version of CART by default. There are many advantages of CART over ID3 or other algorithms like, being faster and more efficient as it is simpler and only uses binary trees. But since CART produces only binary trees, i.e., non leaf nodes will only be splitted into two categories. Thus, our two features need to be One Hot encoded.

One Hot Encoding: Say we have a categorical feature with k classes. Then One Hot encoding it replaces it with k binary features. after doing that, our data looks like:



	AMPHIBIANS (Amphibia)	BIRDS (Aves)	CYADS (Cycadopsida)	FISHES AND AQUATIC SPECIES	FISHES AND OTHER AQUATIC SPECIES	FLOWERING PLANTS (Liliopsida and Magnoliopsida)	HYDROZOANS (Hydrozoa)	INSECTS	MAMMALS (Mammalia)	MOLLUSCS
0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...
2019	0	0	0	0	0	1	0	0	0	0
2020	0	0	0	0	0	1	0	0	0	0
2021	0	0	0	0	0	1	0	0	0	0
2022	0	0	0	0	0	1	0	0	0	0
2023	0	0	0	0	0	1	0	0	0	0

Now, We split the dataset into training and test set so that we can evaluate it later.

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                             test_size = 0.25)
```

Now, We a train random forest classifier on our training set. We use the Gini index criterion for feature selection in the decision nodes.

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 100,
                                    criterion = 'gini')
classifier.fit(X_train, y_train)
```

Predicting the test set results:

```
y_pred = classifier.predict(X_test)
```

Evaluating the results

Now, we evaluate our predictions using confusion matrix and classification metrics like precision recall and f1 score.

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
                 LC   NT   VU   EN   CR   EX
       LC     [[205   10   10   12    3    0]
       NT      [ 35   18    6    4    6    1]
       VU      [ 41    4   11   10    5    0]
       EN      [ 53    5    0   22    0    0]
       CR      [ 27    2    0    3    9    1]
       EX      [  1    0    0    2    0    0]]

              precision    recall  f1-score   support

           0       0.57      0.85      0.68       240
           1       0.46      0.26      0.33        70
           2       0.41      0.15      0.22        71
           3       0.42      0.28      0.33        80
           4       0.39      0.21      0.28        42
           5       0.00      0.00      0.00         3

    accuracy                           0.52       506
   macro avg       0.37      0.29      0.31       506
weighted avg       0.49      0.52      0.48       506
```

Metrics

Here we can see that the accuracy of our model on unseen data is approximately 52%. It means that the model is predicting the status of the species correctly 52% of the time. But, we have to keep in mind that there is a limitation of accuracy as a metric. Suppose we make the given model,

$$\text{model} = \text{argmax}(\text{Response Variable of data set})$$

Then, this model will classify any species into the class with the maximum number of species regardless of what the input parameters were, thus making it a not very ideal metric in our data due to the high difference in number of species in the red list categories.

Precision: Also called positive predictive value, it is measuring the proportion of the species in each red list category that actually belonged in it.

Recall/Sensitivity: It is measuring the proportion of species in each red list category whose status was predicted correctly.

F1- Score: It is the harmonic mean of precision and recall.

Interpretation

From looking at the metrics we can conclude that the previous year's conservation status of a species does affect it for the consecutive year but there are several other factors for example climate change, habitat loss and degradation, environmental contamination and many more that can affect the future conservation status of a species and all of them should be taken into consideration to avoid any species facing a serious extinction crisis.