Can't see the forest for the trees
Decision Trees
& Random Forests

What is a Decision Tree?

A **supervised learning algorithm** used for both classification and regression tasks.

Splits the data into smaller subsets based on feature values, creating a tree-like structure.

Final predictions are made based on the leaf nodes.

Key Feature

Interpretability: Easy to visualize and understand.

https://il.akinator.com/

Entropy

- Measures the uncertainty or impurity in a dataset.

$$E(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

Gini Impurity

- Measures the likelihood of incorrect classification.

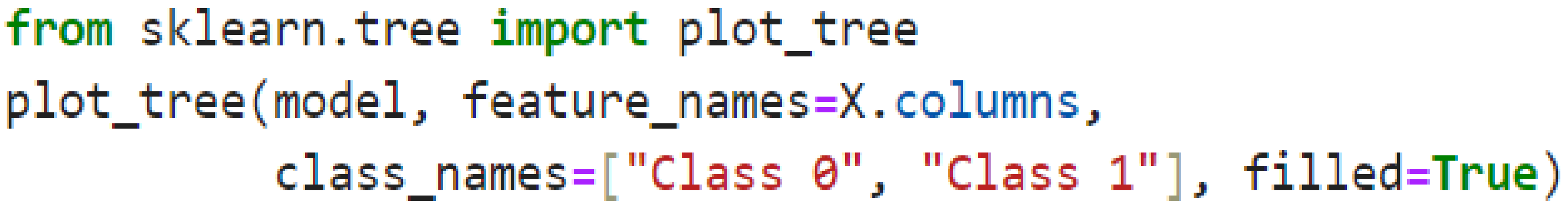
$$G(S) = 1 - \sum_{i=1}^{c} p_i^2$$

Information Gain

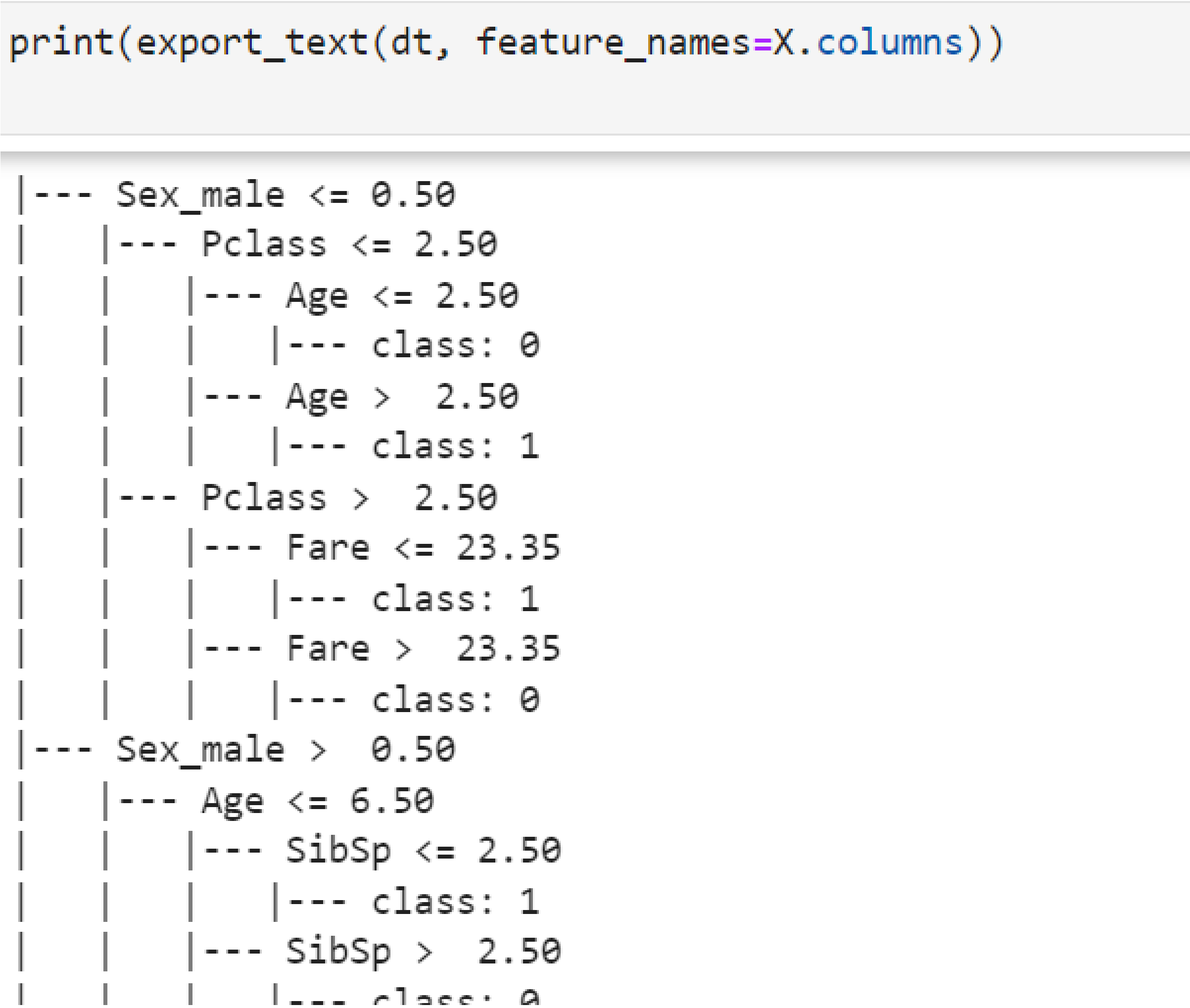
- Measures the reduction in entropy after a split.

Using plot_tree:

Visualizes the structure of a trained decision tree, showing splits, conditions, and leaf nodes.



```
from sklearn.tree import plot_tree
plot_tree(model, feature_names=X.columns,
          class_names=["Class 0", "Class 1"], filled=True)
```



```
print(export_text(dt, feature_names=X.columns))
```

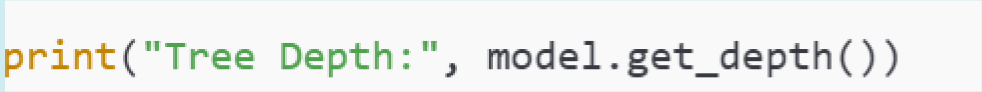
```
|--- Sex_male <= 0.50
|   |--- Pclass <= 2.50
|   |   |--- Age <= 2.50
|   |   |   |--- class: 0
|   |   |--- Age >  2.50
|   |   |   |--- class: 1
|   |--- Pclass >  2.50
|   |   |--- Fare <= 23.35
|   |   |   |--- class: 1
|   |   |--- Fare >  23.35
|   |   |   |--- class: 0
|--- Sex_male >  0.50
|   |--- Age <= 6.50
|   |   |--- SibSp <= 2.50
|   |   |   |--- class: 1
|   |   |--- SibSp >  2.50
|   |   |   |--- class: 0
```

Another way to visualize the tree in a textual way

Decision Tree

get_depth:

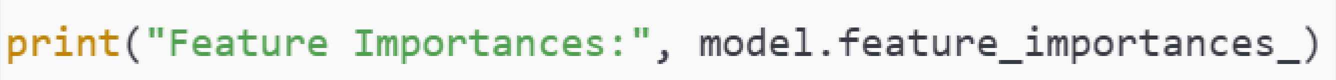
Returns the depth of the trained tree.



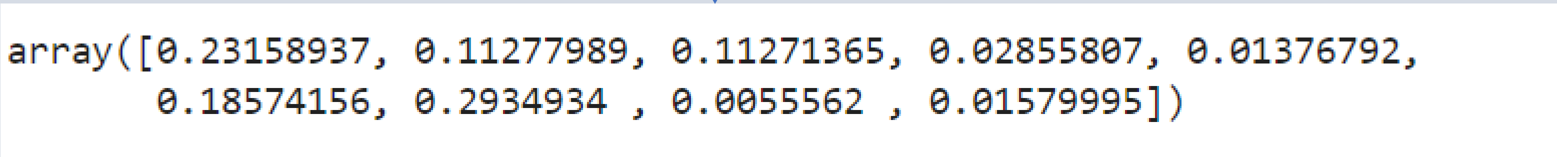
```python
print("Tree Depth:", model.get_depth())
```

feature_importances_:

Indicates the importance of each feature in making predictions.



```python
print("Feature Importances:", model.feature_importances_)
```



```
array([0.23158937, 0.11277989, 0.11271365, 0.02855807, 0.01376792,
       0.18574156, 0.2934934 , 0.0055562 , 0.01579995])
```

The output is an array that can be transformed to a data frame column against the actual features names

What is a Random Forest?

- An **ensemble method** that combines multiple decision trees.
- Uses **Bagging (Bootstrap Aggregation)** to train each tree on a random subset of the data.

Key Features:

- **Robustness:** Reduces overfitting by averaging predictions.
- **Diversity:** Each tree uses random feature subsets for splits.

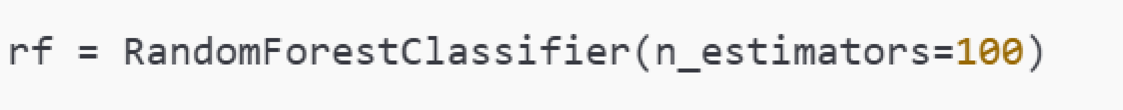
Ensemble Methods Overview

1. Bagging: Combines predictions from multiple models trained on different random subsets of the data to reduce variance.

2. Boosting: Trains models sequentially, where each model focuses on correcting the errors of the previous one, reducing bias.

3. Stacking: Combines the outputs of multiple models (even different types) into a "meta-model" for better predictions.

n_estimators:
The number of trees in the forest.



```
rf = RandomForestClassifier(n_estimators=100)
```

max_depth:
The maximum depth of each tree to prevent overfitting.

max_features:
The maximum number of features considered for each split.

Aspect	Decision Tree	Random Forest
Structure	Single tree	Ensemble of trees
Overfitting	Prone to overfitting	Less prone due to averaging
Interpretability	Easy to interpret	Harder to interpret
Computation Time	Faster	Slower