Machine Learning Lab: Decision Tree and Random Forest Classifier

Nikhil Saraogi [21MCA1080]

Objective

I am using Decision Tree & Random Forest in Predicting the attrition of your valuable employees.

The objective of a decision tree is to make the best decision possible at the end of each node, so it requires an algorithm that can accomplish this. The greedy and recursive algorithm in question is called Hunt's algorithm. Greedy means it makes the best choice possible at each step, and recursive means it breaks the larger question into smaller ones and deals with them in the same way. Purity is a metric used to determine whether to split at each node. When a node's data is evenly split 50/50, it is 100% impure; when all of its data is from a single class, it is 100% pure.

Random Forest is a classifier that uses multiple decision trees on different subsets of the input dataset and averages the results to increase the dataset's predictive accuracy. Instead of relying on a single decision tree, the random forest uses predictions from each tree and predicts the result based on the votes of the majority of predictions. Higher accuracy and over fitting are prevented by the larger number of trees in the forest.

Methodology

Decision Tree is a Supervised Machine Learning Algorithm that uses a set of rules to make decisions, similarly to how humans make decisions.

DecisionTreeClassifier (criterion = 'gini', random_state = None, max_depth = None, min_samples_leaf =1)

Here are a few important parameters:

• criterion: It is used to measure the quality of a split in the decision tree classification. By default, it is 'gini'; it also supports 'entropy'.

• max_depth: This is used to add maximum depth to the decision tree after the tree is expanded.

• min_samples_leaf: This parameter is used to add the minimum number of samples required to be present at a leaf node.

Dataset- IBM HR Analytics Employee Attrition & Performance

Link- https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

Uncover the factors that lead to employee attrition and explore important questions such as ‘show me a breakdown of distance from home by job role and attrition’ or ‘compare average monthly income by education and attrition’. This is a fictional data set created by IBM data scientists.

Education
1 'Below College'
2 'College'
3 'Bachelor'
4 'Master'
5 'Doctor'

EnvironmentSatisfaction
1 'Low'
2 'Medium'
3 'High'
4 'Very High'

JobInvolvement
1 'Low'
2 'Medium'
3 'High'
4 'Very High'

JobSatisfaction
1 'Low'
2 'Medium'
3 'High'
4 'Very High'

PerformanceRating
1 'Low'
2 'Good'
3 'Excellent'
4 'Outstanding'

RelationshipSatisfaction
1 'Low'
2 'Medium'
3 'High'
4 'Very High'

WorkLifeBalance
1 'Bad'
2 'Good'
3 'Better'
4 'Best'

Implementation-

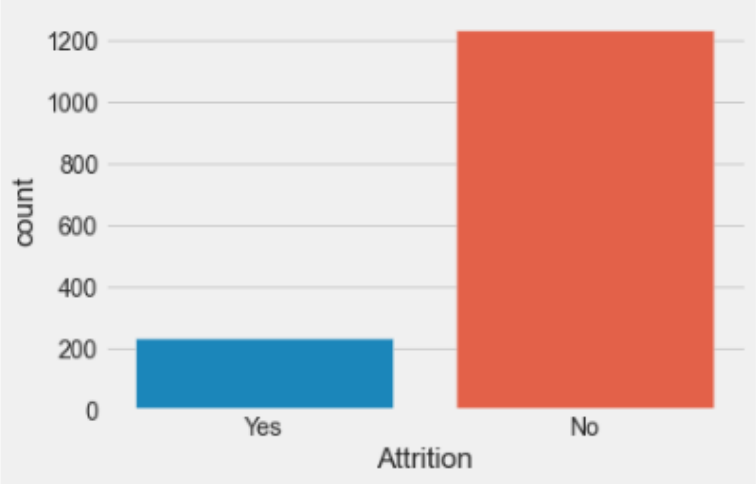
```
df = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
df.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSa
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	

5 rows × 35 columns


```
sns.countplot(x='Attrition', data=df)
```

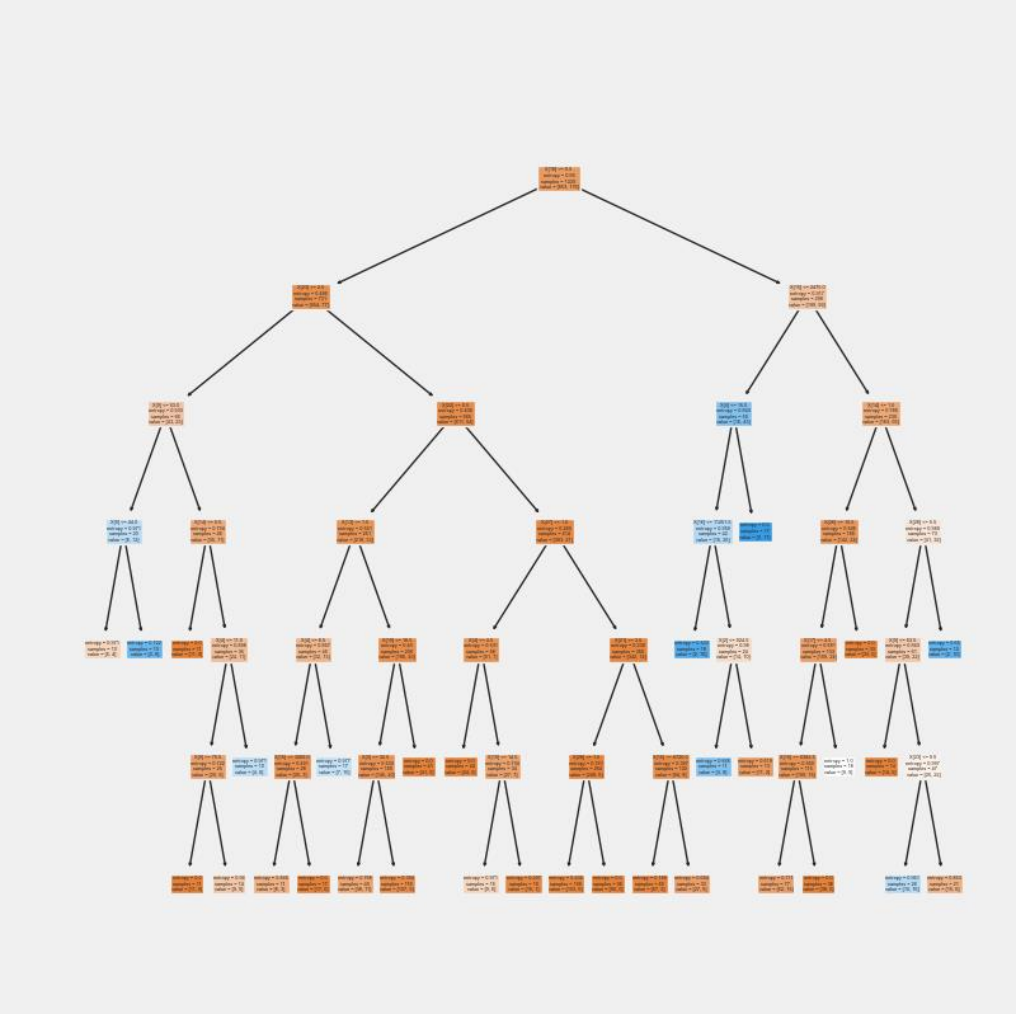
```
<AxesSubplot:xlabel='Attrition', ylabel='count'>
```



Decision Tree-

```
Test Result:
================================================
Accuracy Score: 77.78%

_______________________________________________
CLASSIFICATION REPORT:
                    0           1  accuracy   macro avg  weighted avg
precision    0.887363   0.259740  0.777778    0.573551      0.800549
recall       0.850000   0.327869  0.777778    0.588934      0.777778
f1-score     0.868280   0.289855  0.777778    0.579067      0.788271
support    380.000000  61.000000  0.777778  441.000000    441.000000
_______________________________________________
Confusion Matrix:
 [[323  57]
 [ 41  20]]
```



Random Forest-

```
Test Result:
================================================
Accuracy Score: 86.17%

_______________________________________________
CLASSIFICATION REPORT:
                    0           1  accuracy   macro avg  weighted avg
precision    0.871795    0.500000  0.861678    0.685897      0.820367
recall       0.984211    0.098361  0.861678    0.541286      0.861678
f1-score     0.924598    0.164384  0.861678    0.544491      0.819444
support    380.000000   61.000000  0.861678  441.000000    441.000000
_______________________________________________
Confusion Matrix:
 [[374   6]
 [ 55   6]]
```