IP_4thPart_DecisionTree

December 19, 2023

1 Interview Questions:

1.1 1. Decision Tree

1.2 2. Entropy, Information Gain, Gini Impurity

1.3 3. Decision Tree Working For Categorical and Numerical Features

1.4 4. What are the scenarios where Decision Tree works well

1.5 5. Decision Tree Low Bias And High Variance- Overfitting

1.6 6. Hyperparameter Techniques

1.7 7. Library used for constructing decision tree

1.8 8. Impact of Outliers Of Decision Tree

1.9 9. Impact of mising values on Decision Tree

1.10 10. Does Decision Tree require Feature Scaling

2 Decision Tree Classifier And Regressor

3 what is decison Tree?

```
Decision Tree is ML algorithm used for both classification and regression
algorithm will look like Tree based struture thats why its called decision tree.
```

4 Decision Tree components are :

```
1. root node -The most important feature (more purity than all)
2. decision nodes -(next important features or next features which has better purity)
3. leaf nodes -output values
```

5 how decision tree works?

```
1.decision tree works based on split the data's into subsets based on its features.
2.features will placed in a form of nodes.
3.we will stop the split at the subsets falls into 100% on one class or if its reaches it maxi
4.if no.of examples are lesser than the threshold
```

6 what is purity

how efficiently one feature can split the datasets into subsets based on the output class.
ex: imagine datasets contain 5 dogs,5 cats
features are equal to face shape,ear shape.
based on face shape we can split the datasets into 5 dogs =0/5 is a cat and 5 cats 5/5 cats =1

7 what is entropy?

Measurement of impurity or purity in a dataset is called entropy.
its values start from 0 go along 1 and comes at 0.
Based on previous cat classfication example:
if one feature can split 4/5 is a cat,then we can say entropy =0.9

8 what is information gain ?

reduction of entropy is called information gain

8.1 why we need information gain?

we cant decide the good purity feature by only its entropy ,because
high entropy created by its large datasets is worse than high entropy created by small subsets
so,we should give importance for no.of subsets in that entropy too.
thats why we have information gain ,used to find the value which consider both entropy and no.
its called information gain.

information gain=H(p_root)-(h(p)*no.of true subsets /total_no_of_subsets)

9 what are all decision points?

1.How to choose root node feature?
    1.Based on the feature purity ,we will choose root node

2.When to stop the split?(depth,information gain threshold,subsets threshold are hyperparamate
    1.When feature can split the subsets 100% on one class (100% purity)
    2.When depth exceeds its max threshold
    3.when information gain is lesser than threshold for one feature
    4.when we have less data than the threshold for split

10 what will you do if feature contain more than two category?

we can do one hot encoding ,so that no.of categories will convert as columns.then we can use t

11 what will you do if feature contains discrete values instead of continous values?

```
we will find the threshold value which can give good entropy.then we will use that threshold a
```

12 how it will work for regression?

```
1.It will work based on variance instead of entropy.
2.variance -how one point differes from its mean
3.we will do redcuction of variance instead of reduction of entropy to find the feature.
4.we will make the decision by its avg of the final split datasets.
```

13 2. Advantages

Advantages of Decision Tree

1. Clear Visualization: The algorithm is simple to understand, interpret and visualize as the idea is mostly used in our daily lives. Output of a Decision Tree can be easily interpreted by humans.
2. Simple and easy to understand: Decision Tree looks like simple if-else statements which are very easy to understand.
3. Decision Tree can be used for both classification and regression problems.
4. Decision Tree can handle both continuous and categorical variables.
5. No feature scaling required: No feature scaling (standardization and normalization) required in case of Decision Tree as it uses rule based approach instead of distance calculation.
6. Handles non-linear parameters efficiently: Non linear parameters don't affect the performance of a Decision Tree unlike curve based algorithms. So, if there is high non-linearity between the independent variables, Decision Trees may outperform as compared to other curve based algorithms.
7. Decision Tree can automatically handle missing values.
8. Decision Tree is usually robust to outliers and can handle them automatically.
9. Less Training Period: Training period is less as compared to Random Forest because it generates only one tree unlike forest of trees in the Random Forest.

14 disadvantages

```
1.more sensitive to small changes in dataset.
2.overfitting - it will train very well with train dataset.
3.variance -but for new dataset that algoithm will struggle to give the result.thats why we ha
4.Not suitable for large datasets
```

15 application

Fraud Detection: Identifying potentially fraudulent transactions based on decision tree analysi

[1]:
```
# coding example
```

[10]:
```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
from sklearn import tree
```

[5]:
```
X=load_iris().data
```

[6]:
```
y=load_iris().target
```

[11]:
```
cls=DecisionTreeClassifier()
```

[12]:
```
cls.fit(X,y)
```

[12]:
```
DecisionTreeClassifier()
```

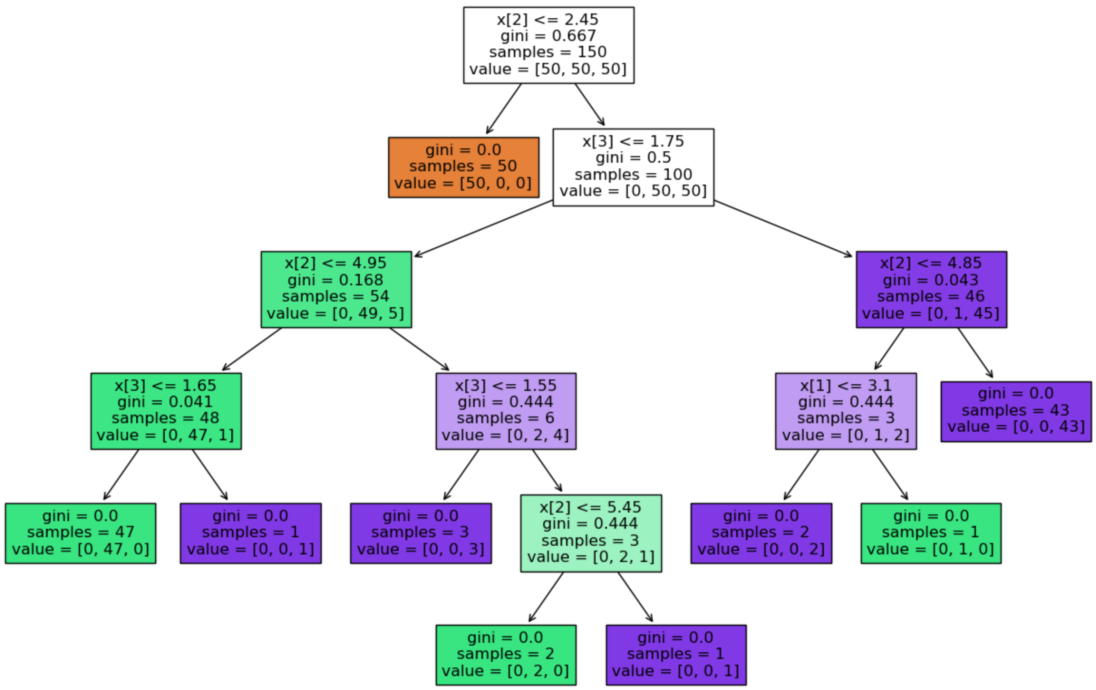
[13]:
```
cls.score(X,y)
```

[13]:
```
1.0
```

[16]:
```
plt.figure(figsize=(15,10))
tree.plot_tree(cls,filled=True)
```

[16]:
```
[Text(0.5, 0.9166666666666666, 'x[2] <= 2.45\ngini = 0.667\nsamples = 150\nvalue
= [50, 50, 50]'),
 Text(0.4230769230769231, 0.75, 'gini = 0.0\nsamples = 50\nvalue = [50, 0, 0]'),
 Text(0.5769230769230769, 0.75, 'x[3] <= 1.75\ngini = 0.5\nsamples = 100\nvalue
= [0, 50, 50]'),
 Text(0.3076923076923077, 0.5833333333333334, 'x[2] <= 4.95\ngini =
0.168\nsamples = 54\nvalue = [0, 49, 5]'),
 Text(0.15384615384615385, 0.4166666666666667, 'x[3] <= 1.65\ngini =
0.041\nsamples = 48\nvalue = [0, 47, 1]'),
 Text(0.07692307692307693, 0.25, 'gini = 0.0\nsamples = 47\nvalue = [0, 47,
0]'),
 Text(0.23076923076923078, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(0.46153846153846156, 0.4166666666666667, 'x[3] <= 1.55\ngini =
0.444\nsamples = 6\nvalue = [0, 2, 4]'),
 Text(0.38461538461538464, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
 Text(0.5384615384615384, 0.25, 'x[2] <= 5.45\ngini = 0.444\nsamples = 3\nvalue
= [0, 2, 1]'),
 Text(0.46153846153846156, 0.08333333333333333, 'gini = 0.0\nsamples = 2\nvalue
= [0, 2, 0]'),
 Text(0.6153846153846154, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
```

```
[0, 0, 1]'),
 Text(0.8461538461538461, 0.5833333333333334, 'x[2] <= 4.85\ngini =
0.043\nsamples = 46\nvalue = [0, 1, 45]'),
 Text(0.7692307692307693, 0.4166666666666667, 'x[1] <= 3.1\ngini =
0.444\nsamples = 3\nvalue = [0, 1, 2]'),
 Text(0.6923076923076923, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
 Text(0.8461538461538461, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
 Text(0.9230769230769231, 0.4166666666666667, 'gini = 0.0\nsamples = 43\nvalue =
[0, 0, 43]')]
```



[18]:
```
tree.export_text(cls)
```

[18]:
```
'|--- feature_2 <= 2.45\n|   |--- class: 0\n|--- feature_2 >  2.45\n|   |---
feature_3 <= 1.75\n|   |   |--- feature_2 <= 4.95\n|   |   |   |--- feature_3 <=
1.65\n|   |   |   |   |--- class: 1\n|   |   |   |--- feature_3 >  1.65\n|   |
|   |   |   |--- class: 2\n|   |   |--- feature_2 >  4.95\n|   |   |   |---
feature_3 <= 1.55\n|   |   |   |   |--- class: 2\n|   |   |   |--- feature_3 >
1.55\n|   |   |   |   |--- feature_2 <= 5.45\n|   |   |   |   |   |--- class:
1\n|   |   |   |   |--- feature_2 >  5.45\n|   |   |   |   |   |--- class: 2\n|
|--- feature_3 >  1.75\n|   |   |--- feature_2 <= 4.85\n|   |   |   |---
feature_1 <= 3.10\n|   |   |   |   |--- class: 2\n|   |   |   |--- feature_1 >
3.10\n|   |   |   |   |--- class: 1\n|   |   |--- feature_2 >  4.85\n|   |   |
|--- class: 2\n'
```

[ ]:

1. What Are the Basic Assumption? There are no such assumptions

[ ]:

4. Whether Feature Scaling is required? No

6. Impact of outliers? It is not sensitive to outliers.Since, extreme values or outliers, never cause much reduction in RSS, they are never involved in split. Hence, tree based methods are insensitive to outliers.

[ ]:

Types of Problems it can solve(Supervised)

1. Classification
2. Regression

[ ]:

Overfitting And Underfitting Ho to avoid overfitting

https://www.youtube.com/watch?v=SLOyyFHbiqo

[ ]:

16 Gini Impurity:

```
1.we can find the impurity by this method instead of entropy,
2.But,its computationly faster than entropy function because it doesnt have any log conversions
3.It starts from 0 to 0.5 and again it comes to the 0
```

[ ]:

17 Cost Complexity pruning :

```
This methods used to select no.of depths which can produce good amount of accuracy level by fi
by trying different value of depths we can find good alpha which can give good accuracy by try
alphas ordepths.
```

18 References:

18.1 1. Tutorial 37:Entropy In Decision Tree https://www.youtube.com/watch?v=1IQOtJ4NI_0

18.2 2. Tutorial 38:Information Gain https://www.youtube.com/watch?v=FuTRucXB9rA

18.3 3. Tutorial 39:Gini Impurity https://www.youtube.com/watch?v=5aIFgrrTqOw

18.4 4. Tutorial 40: Decision Tree For Numerical Features: https://www.youtube.com/watch?v=5O8HvA9pMew

18.5 5. How To Visualize DT: https://www.youtube.com/watch?v=ot75kOmpYjI

18.6 6. Decision Trees and Ensemble Methods | Stanford CS229: Machine Learning https://www.youtube.com/watch?v=wr9gUr-eWdA&list=PLoROMvodv4rMiGQp3WXShtMGgzqpfVfbU&index=10

18.7 7. code basics https://www.youtube.com/watch?v=PHxYNGo8NcI