← Go Back to Machine Learning

Course Content

FAQs - Decision Tree

1. When should we use Precision or Recall as model performance evaluation criteria?

Precision should be used when you want to minimize False Positives i.e. one wants at least negatives should not be predicted as positives. Also, in cases where the loss of resources is high.

Recall should be used when you want to minimize False Negatives, i.e. one wants at least positives should not be predicted as negatives. Also, in cases where the loss of opportunity is high.

2. What is misclassification?

Misclassification occurs when values are predicted incorrectly or the model assigns the observation to a different class instead of the class it should be in. For example, for observation, the actual label is class 0 but the model predicts this observation as class 1.

3. Why confusion matrix is inverted in the hands-on notebook and it is difficult to identify TP, TN, FP, and FN?

Generally, in theory, or while teaching confusion matrix many sources keep the predicted labels on the y-axis whereas the actual labels are on the x-axis.

But in the implementation, it can be different depending upon the library we are using.
Sklearn follows an approach of Actual labels on the x-axis and Predicted labels on the y-axis.

Let us understand how to identify TP, TN, FP, and FN in a confusion matrix through an example:

Let's say we are trying to predict whether a person is diabetic (class 1) or not (class 0).

- True Positives (TP): A person has diabetes and the model predicts that person is diabetic.
- True Negatives (TN): A person doesn't have diabetes and the model predicted that person doesn't have diabetes.
- False Positives (FP): A person doesn't have diabetes but the model predicted that person has diabetes.
- False Negatives (FN): The person has diabetes but the model predicted that person doesn't have diabetes.

Now based on the actual label and predicted label you can identify the TP, TN, FP, and FN.

4. Which evaluation metric should be selected for which scenario?

The evaluation metrics hold different importance in different problem scenarios.

. When we want to reduce the False Negatives in our model, we try to improve the Recall.

When we want to reduce the False Positives in our model, we try to improve the Precision.

When we want both FP and FN to reduce, we try to increase the F1-Score.

When we want a more generalized model with accurate predictions in terms of TP and TN, we increase the accuracy.

5. Decision Tree arrows are missing, how to fix this?

Use the following code as your reference to resolve the issue and make necessary changes in the name of the model, feature names, etc.

```
plt.figure(figsize=(20,30))
out = tree.plot_tree(model,feature_names=feature_names,filled=True,fontsize=9,node_ids=F
#below code will add arrows to the decision tree split if they are missing
for o in out:
     arrow = o.arrow_patch
     if arrow is not None:
        arrow.set_edgecolor('black')
        arrow.set_linewidth(1)
plt.show()
```

6. What is the impact of alpha on a decision tree?

Ans: Alpha is the complexity parameter and accounts for how much error is there in the model after pruning the tree.

alpha = (error in pruned tree - error in original tree)/no. of nodes reduced

The error in the pruned tree will always be greater than the error in the original tree. This is because, the original tree is fully grown and will overfit the data with very low or negligible error, whereas the pruned tree has its branches pruned and thus will have a higher error than the original tree.

Now, when alpha is really high, it means that the error(pruned tree) >> error(original tree), and hence the pruned part plays important in determining the result. When alpha is very low, it means that error(pruned tree) ~ error(original tree). They have almost the same errors.

7. When do we label encode and create dummy variables for categorical levels?

We generally prefer label encoding when there is a sense of order on the values, for example, let's say a variable has values bad, good, very good in such a case we know that there is an order and we can encode them as 1,2,3 respectively.

But let's say the values are red, blue, green in this case there is no definite order in values and hence creating dummy variables would be a better choice.

❮ Previous

Next ❯