Loss Function

quantra

Objective:

In this document, we will cover a couple of important loss functions that are used in decision tree models.

Loss function in Machine learning

A loss function is a metric which is used to evaluate the performance of a machine learning algorithm. A machine learning algorithm is usually first trained on a portion of the data and then tested for its accuracy on the remaining data. The loss function reflects the success of the model in making predictions on this unseen data. Based on the value of the loss function, one can fine tune the model to get the desired accuracy and thereby know the viability of the machine learning model.

How to know which loss function is to be used for a machine learning model?

There are a number of loss functions which are available for different machine learning algorithms. The choice of the right loss function is dependent on the problem that one is trying to solve using the machine algorithm. For example, decision trees can be used for classification or regression problems.

Accordingly, for decision tree models we have loss functions like gini impurity, entropy for classification problems and loss functions like mean square error (MSE) and mean absolute error (MAE) for regression problems. One can also write a custom loss function depending on the problem statement.

Gini Impurity

Gini impurity is used for classification tree models. Gini impurity computes the impurity of the dataset at every node for the available features and to what extent impurity is reduced if the internal node is split using a particular feature. The objective is to have a more homogeneous data after the split. Gini impurity is given by the equation seen given below.

$$Gini = 1 - \sum_j p_j^2$$

In the equation, pj represents the probability of class j. The gini impurity is calculated by subtracting the sum of the squared probabilities of each class from one.

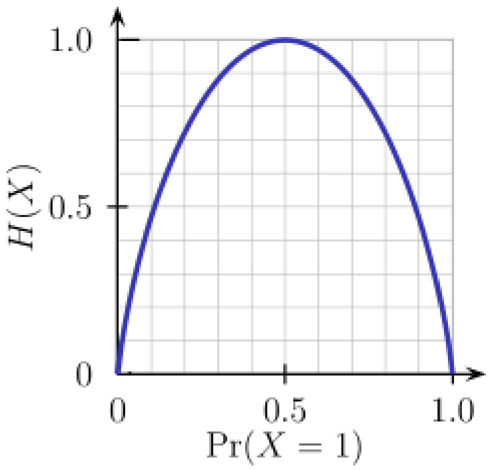
You can learn more about gini impurity in the 'Splitting Measures' video where we have covered it with a detailed example.

Entropy and Information Gain

The entropy measure is similar to gini impurity except that it includes log function and hence can be computationally intensive. Entropy is given by the equation seen given below. In this equation, p_j is the probability of class j.

$$Entropy = -\sum_j p_j \log_2 p_j$$

Entropy lies between 0 and 1. If the sample dataset is completely homogeneous, the entropy is zero. If the sample is equally divided then its entropy is one.



The aim of the algorithm would be to decrease the entropy. It applies the same logic as explained in gini impurity. In case of entropy, it computes the information gain for each feature and splits the tree accordingly.

Mean Absolute Error (MAE)

The mean absolute error (MAE) is used for regression trees and it computes the average magnitude of errors in predictions. While computing the errors, MAE ignores the direction of the errors. It takes the absolute value of the errors and averages it out over the number of observations. MAE is given by the equation seen given below.

$$MAE = \frac{1}{n}\sum_{i}^{n} |y_i - \hat{y}_i|$$

In the equation above, y_i contains the predicted values and $\hat{y}_i$ contain the actual values.

Mean Square Error (MSE)

Mean squared error (MSE) is calculated as given in the equation below. It finds the average squared distance between the predicted values and the actual values. The squaring is done so that negative values do not cancel out the positive values.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Specifying Loss function in Scikit-learn

In the scikit-learn Python library, the loss function can be specified when defining a classifier. For example, in case of a classification tree model, we can specify either ‘gini’ or ‘entropy’ as the loss function in the decision tree classifier, ‘DecisionTreeClassifier’.

The Python code for a DecisionTreeClassifier is shown below.

```
from sklearn.tree import DecisionTreeClassifier
clf= DecisionTreeClassifier()
clf

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```

View DecisionTreeClassifier Parameters

One can use the Python help function on DecisionTreeClassifier to view the all the parameters. The snippet below shows the description of the 'criterion' parameter of the DecisionTreeClassifier.

```
help(DecisionTreeClassifier)


Parameters
----------
criterion : string, optional (default="gini")
    The function to measure the quality of a split. Supported criteria are
    "gini" for the Gini impurity and "entropy" for the information gain.
```

As can be seen, the scikit-learn classification tree classifier supports two criterion namely, gini and entropy which are used as loss functions in the decision tree model. These are used by the model for growing the tree and it the decision tree tries to ensure the best split at every internal node of the model based on the selected criterion.

Summary

Decision tree models mainly use loss functions like gini impurity, entropy, and RMSE. One can try running a decision tree model using different loss functions and see their behaviour on the model accuracy.