Decision tree and CV Assignment:

1) How decision tree works for a regression problem?

Regression trees are used when the dependent variable is continuous.  
For regression trees, the value of terminal nodes is the mean of the observations falling in that region. Therefore, if an unseen data point falls in that region, we predict using the mean value.

2) Why Recursive binary splitting is called Greedy Approach?

**Recursive Binary Splitting**

In this procedure all the features are considered and different split points are tried and tested using a cost function. The split with the best cost (or lowest cost) is selected.

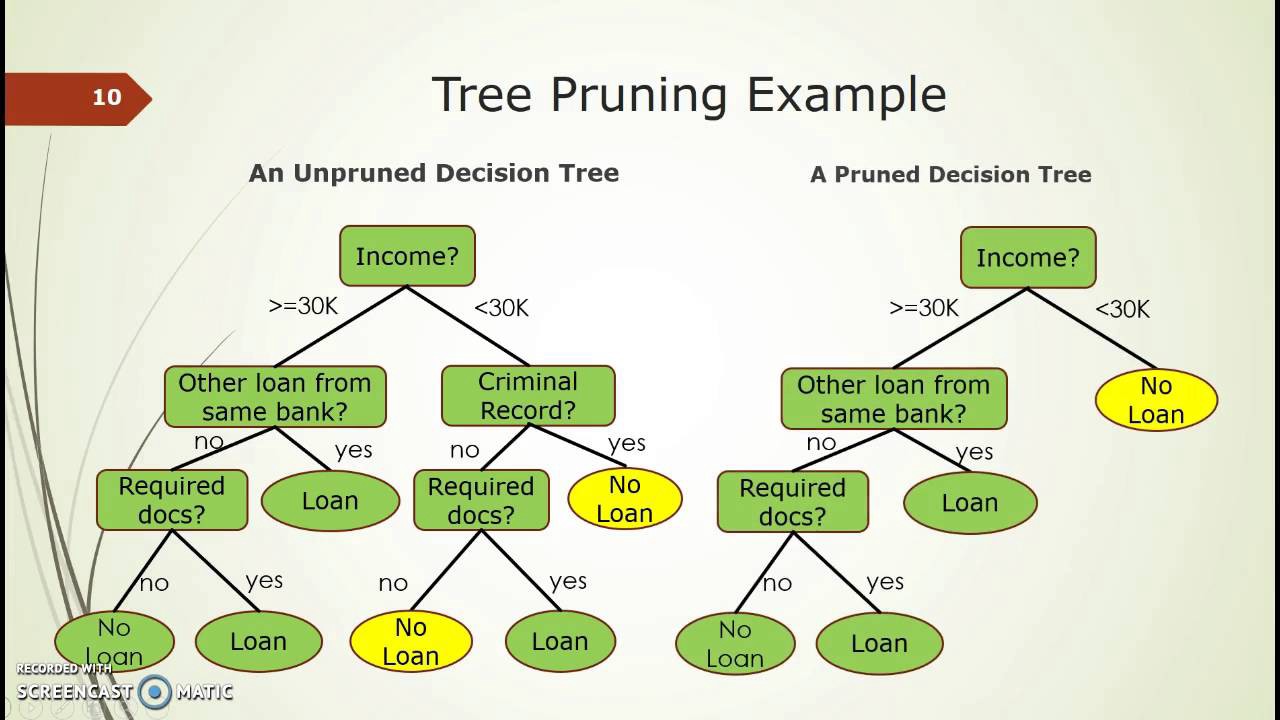
This **algorithm is recursive in nature** as the groups formed can be sub-divided using same strategy. Due to this procedure, this algorithm is also known as the **greedy algorithm**, as we have an excessive desire of lowering the cost. **This makes the root node as best predictor/classifier.**

3) What do you understand by Greedy approach?

Both the trees follow a top-down greedy approach known as recursive binary splitting. We call it as ‘top-down’ because it begins from the top of tree when all the observations are available in a single region and successively splits the predictor space into two new branches down the tree. It is known as ‘greedy’ because, the algorithm cares (looks for best variable available) about only the current split, and not about future splits which will lead to a better tree.

4) What is Pruning?

The shortening of branches of the tree. Pruning is the process of reducing the size of the tree by turning some branch nodes into leaf nodes, and removing the leaf nodes under the original branch. Pruning is useful because classification trees may fit the training data well, but may do a poor job of classifying new values. A simpler tree often avoids over-fitting.



5) What’s the difference between pre pruning and post pruning?

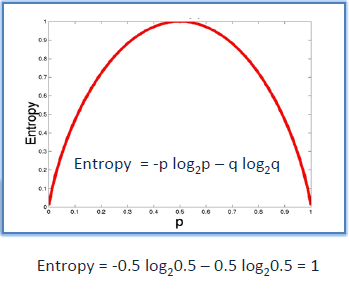
to prevent overfitting is to try and stop the tree-building process early, before it produces leaves with very small samples. This heuristic is known as early stopping but is also sometimes known as pre-pruning decision trees.

* **Pre-pruning** that stop growing the tree earlier, before it perfectly classifies the training set.
* **Post-pruning** that allows the tree to perfectly classify the training set, and then post prune the tree.

<https://www.saedsayad.com/decision_tree_overfitting.htm>

6) What is Entropy? How is it calculated?

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogeneous). ID 3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

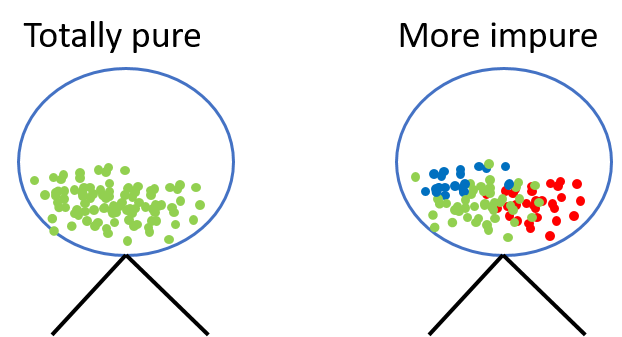


|  |
| --- |
| DefcalculateEntropy(dataSet): |
|  | number = len(dataSet) |
|  | labelCounts = {} |
|  | for featureVector in dataSet: |
|  | currentLabel = featureVector[-1] |
|  | if currentLabel not in labelCounts.keys(): |
|  | labelCounts[currentLabel] = 0 |
|  | labelCounts[currentLabel] +=1 |
|  | entropy = 0 |
|  | for i in labelCounts: |
|  | probability = float(labelCounts[keys])/number |
|  | entropy -=probability\*log(probability,2) |
|  | return entropy |

7) What is Gini Impurity?

Pure means, in a selected sample of dataset all data belongs to same class (PURE).

Impure means, data is mixture of different classes.



Gini Impurity is a measurement of the likelihood of an incorrect classification of a new instance of a random variable, if that new instance were randomly classified according to the distribution of class labels from the data set.If our dataset is Pure then likelihood of incorrect classification is 0. If our sample is mixture of different classes then likelihood of incorrect classification will be high.

*Information Gain = Entropy(parent node) — [Avg Entropy(children)]*

**Steps to calculate entropy for a split:**

1. Calculate the entropy of the parent node
2. Calculate entropy of each individual node of split and calculate the weighted average of all sub-nodes available in a split.

8) What do you understand by Information Gain? How does it help in tree building?

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

1. Calculate entropy of the target.
2. The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain, or decrease in entropy.
3. Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

9) How does node selection take place while building a tree?

1. Both the trees divide the predictor space (independent variables) into distinct and non-overlapping regions. For the sake of simplicity, you can think of these regions as high dimensional boxes or boxes.
2. Both the trees follow a top-down greedy approach known as recursive binary splitting. We call it as ‘top-down’ because it begins from the top of tree when all the observations are available in a single region and successively splits the predictor space into two new branches down the tree. It is known as ‘greedy’ because, the algorithm cares (looks for best variable available) about only the current split, and not about future splits which will lead to a better tree.
3. This splitting process is continued until a user defined stopping criteria is reached. For example: we can tell the the algorithm to stop once the number of observations per node becomes less than 50.
4. In both the cases, the splitting process results in fully grown trees until the stopping criteria is reached.

10) What are different algorithms available for decision tree?

* ID3 (Iterative Dichotomiser) : It is one of the algorithms used to construct decision tree for classification. It uses Information gain as the criteria for finding the root nodes and splitting them. It only accepts categorical attributes.
* C4.5 : It is an extension of ID3 algorithm, and better than ID3 as it deals both continuous and discreet values.It is also used for classfication purposes.
* Classfication and Regression Algorithm(CART) : It is the most popular algorithm used for constructing decison trees. It uses ginni impurity as the default calculation for selecting root nodes, however one can use "entropy" for criteria as well. This algorithm works on both regression as well as classfication problems. We will use this algorithm in our pyhton implementation.

11) What’s the main difference between Gini Impurity and Entropy on the basis of computation time?

Entropy and Ginni impurity can be used reversibly. It doesn't affects the result much. Although, ginni is easier to compute than entropy, since entropy has a log term calculation. That's why CART algorithm uses ginni as the default algorithm.

If we plot ginni vs entropy graph, we can see there is not much difference between them:

12) What are the disadvantages and advantages of using a Decision Tree?

##### *Advantages of Decision Tree:*

* It can be used for both Regression and Classification problems.
* Decision Trees are very easy to grasp as the rules of splitting is clearly mentioned.
* Complex decision tree models are very simple when visualized. It can be understood just by visualising.
* Scaling and normalization are not needed.

##### *Disadvantages of Decision Tree:*

* A small change in data can cause instability in the model because of the greedy approach.
* Probability of overfitting is very high for Decision Trees.
* It takes more time to train a decision tree model than other classification algorithms.

13) How do you deploy model in Heroku?

14) What challenges you faced while deploying the model?

Nothing

Cross Validation

1) What is Cross validation?

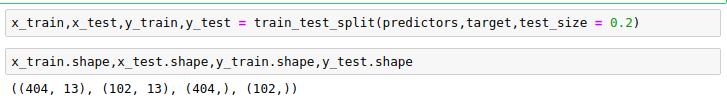
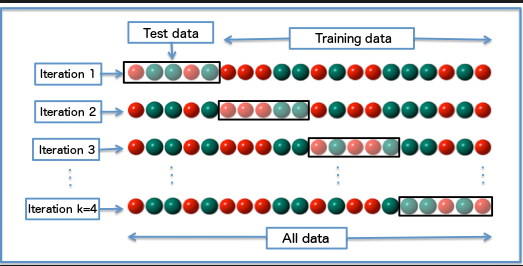
Cross validation (CV) is one of the technique used to test the effectiveness of a machine learning models, it is also a re-sampling procedure used to evaluate a model if we have a limited data.

Cross Validation is a technique which involves reserving a particular sample of a dataset on which you do not train the model. Later, you test your model on this sample before finalizing it.

2) Why do we need to implement Cross validation?

To evaluate the performance of any machine learning model we need to test it on some unseen data. Based on the models performance on unseen data we can say weather our model is Under-fitting/Over-fitting/Well generalised. Cross validation (CV) is one of the technique used to test the effectiveness of a machine learning models, it is also a re-sampling procedure used to evaluate a model if we have a limited data. To perform CV we need to keep aside a sample/portion of the data on which is do not use to train the model, later us this sample for testing/validating.

3) What are different types of CV methods?

1. **Train\_Test Split approach (**ideally split the data into 70:30 or 80:20.) use the **train\_test\_split** of scikit-learn method for this task. 
2. **K-Folds Cross Validation**: K-Fold is a popular and easy to understand, it generally results in a less biased model compare to other methods. Because it ensures that every observation from the original dataset has the chance of appearing in training and test set. This is one among the best approach if we have a limited input data. This method follows the below steps. 
3. **Leave one out cross validation (LOOCV) -** In this approach, we reserve only one data point from the available dataset, and train the model on the rest of the data. This process iterates for each data point.

### Stratified k-fold cross validation - Stratification is the process of rearranging the data so as to ensure that each fold is a good representative of the whole. For example, in a binary classification problem where each class comprises of 50% of the data, it is best to arrange the data such that in every fold, each class comprises of about half the instances. https://cdn.analyticsvidhya.com/wp-content/uploads/2015/11/skfold.png

### <https://www.analyticsvidhya.com/blog/2018/05/improve-model-performance-cross-validation-in-python-r/>

4) How bias and variance varies for each CV method?

When you perform k-fold CV, you get k different estimates of your model’s error- say e_1, e_2, e_3, ..., e_k. Since each e_i is an error estimate, it should ideally be zero.

To check out you model’s ***bias***, find out the **mean**of all the e_is. If this value is low, it basically means that your model gives low error on an average– indirectly ensuring that your model’s notions about the data are accurate enough.

To check out your model’s **variance**, compute the ***standard deviation*** of all the e_is. If this value is high, it means that your model’s performance(in terms of its error) varies a lot with the dataset used for training- something you don’t want.

Obviously, what is ‘high’ enough to rule out your model as being ineffective, depends on your particular application and your domain knowledge (and subsequently your skill as a data miner). Nevertheless, k-fold Cross-Validation is a powerful practical tool to quantify a model’s performance, and has been shown to pretty accurately estimate the real-world numbers in a wide range of scenarios.

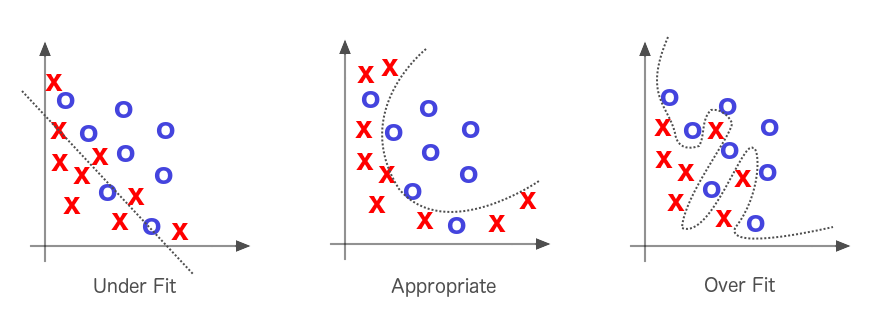
5) Is Train Test Split a kind of CV? True or False.

True

6) How can we check over fitting using CV?

cross-validation does not prevent overfitting in itself, but it may help in identifying a case of overfitting.”

If the model has a good fit on train samples but a poor fit to test samples,  you know it's overfitting, and should use a less flexible model.



<https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85>