MACHINE LEARNING

1211635

MAHASARAKHAM
UNIVERSITY

DECISION TREE

Tree based modeling

Olarik Surinta, PhD.
Lecturer

What is a decision tree?

- Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) this is mostly used in classification problems.
- It works for both categorical and continuous input and output variables.
-

Cr: https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

What is a decision tree?

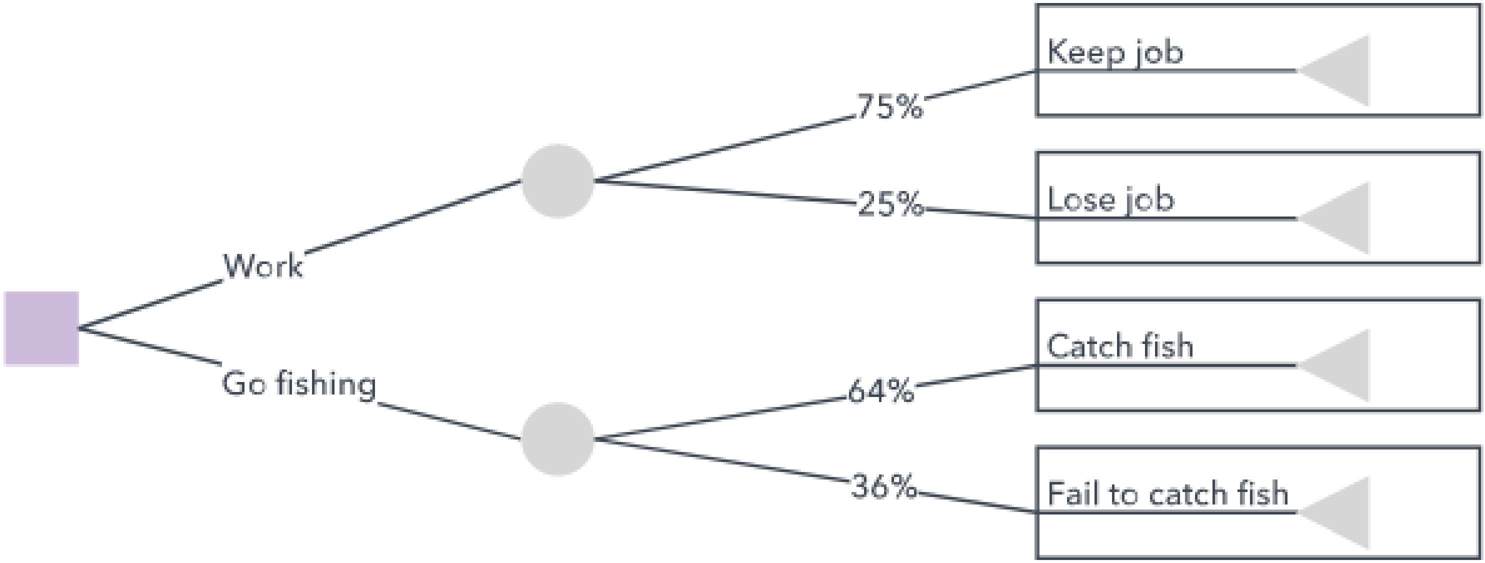
- A decision tree typically starts with a **single node**, which branches into possible outcomes.
- Each of those outcomes leads to *additional nodes*, which branch off into other possibilities.
- This gives it a treelike shape.

Cr: https://www.lucidchart.com/pages/decision-tree

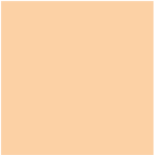
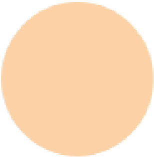
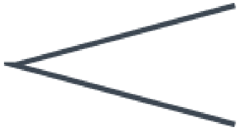
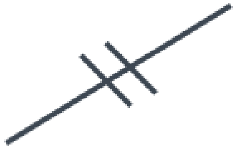
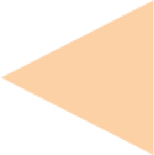
What is a decision tree?

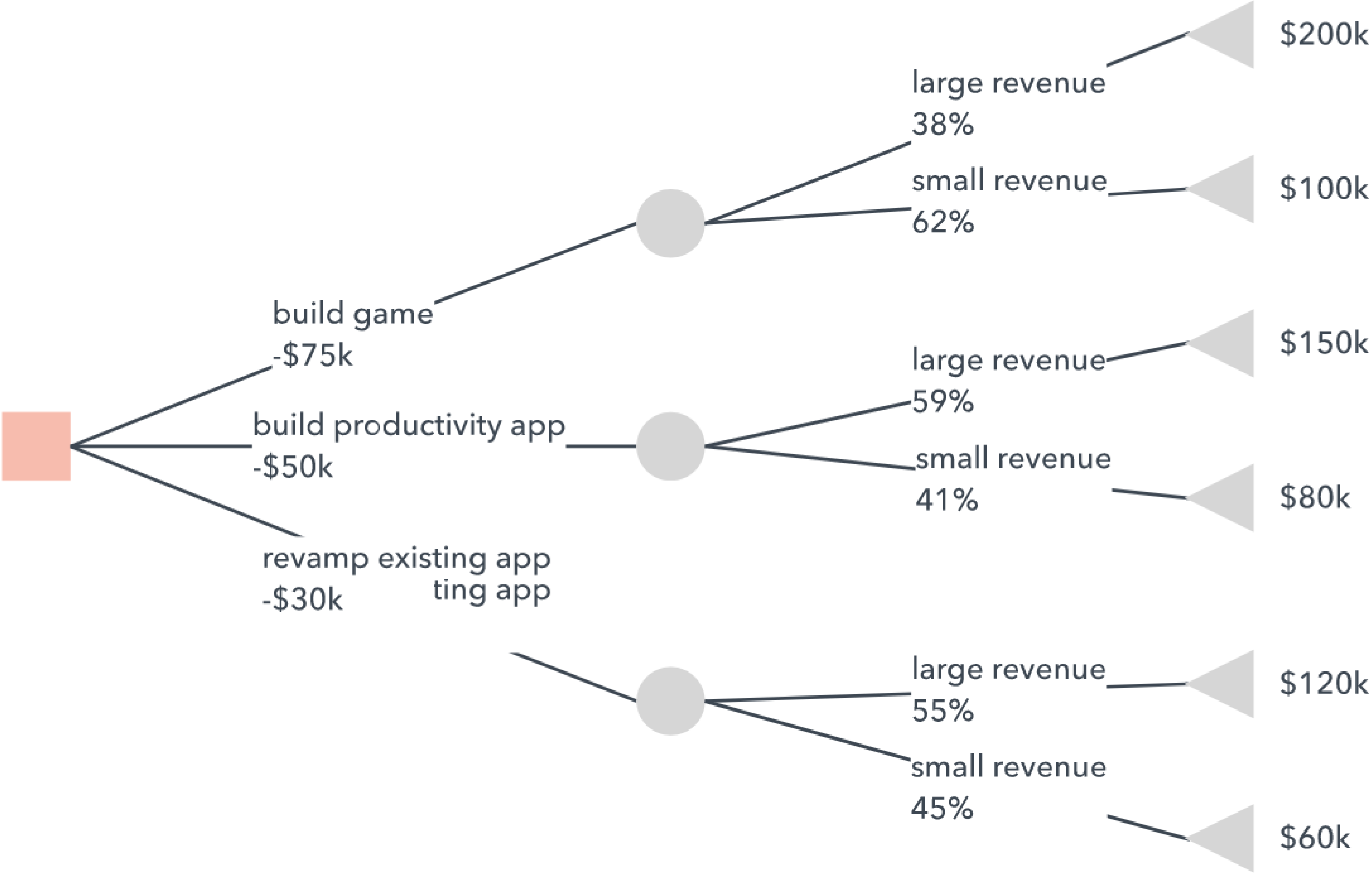
- There are three different types of nodes:
  - **Chance nodes**, represented by a circle, shows the probabilities of certain results.
  - **Decision nodes**, represented by a square, shows a decision to be made.
  - **End nodes** shows the final outcome of a decision path.

What is a decision tree?



Decision tree symbols

Shape	Name	Meaning
	Decision node	Indicates a decision to be made
	Chance node	Shows multiple uncertain outcomes
	Alternative branches	Each branch indicates a possible outcome or action
	Rejected alternative	Shows a choice that was not selected
	Endpoint node	Indicates a final outcome


$200k
large revenue
38%
small revenue
62%
$100k
build game
-$75k
$150k
large revenue
59%
build productivity app
-$50k
small revenue
41%
$80k
revamp existing app
-$30k
ting app
large revenue
55%
$120k
small revenue
45%
$60k

Decision trees in machine learning and data mining

- A decision tree can also be used to help build automated predictive models, which have applications in machine learning, data mining, and statistics.
- Known as decision tree learning, this method takes into account observations about an item to predict that item’s value.

Decision trees in machine learning and data mining

- In these decision trees, nodes represent data rather than decisions.
- This type of three is also known as a classification tree.
- Each branch contains a set of attributes, or classification rules, that are associated with a particular class label, which is found at the end of the brance.

Decision trees in machine learning and data mining

- These rules, also known as decision rules, can be expressed in an if-then clause, with each decision or data value forming a clause, such that, for instance, “if conditions 1, 2, and 3 are fulfilled, then outcome x will be the result with y certainty.”

Decision trees in machine learning and data mining

- Sometimes the predicted variable will be a real number, such as a price.
- Decision trees with continuous, infinite possible outcomes are called regression trees.

Decision trees in machine learning and data mining

- For increased accuracy, sometimes multiple trees are used together in ensemble methods:
  - **Bagging** creates multiple trees by resampling the source data, then has those trees vote to reach consensus.
  - **A Random Forest classifier** consists of multiple trees designed to increased the classification rate

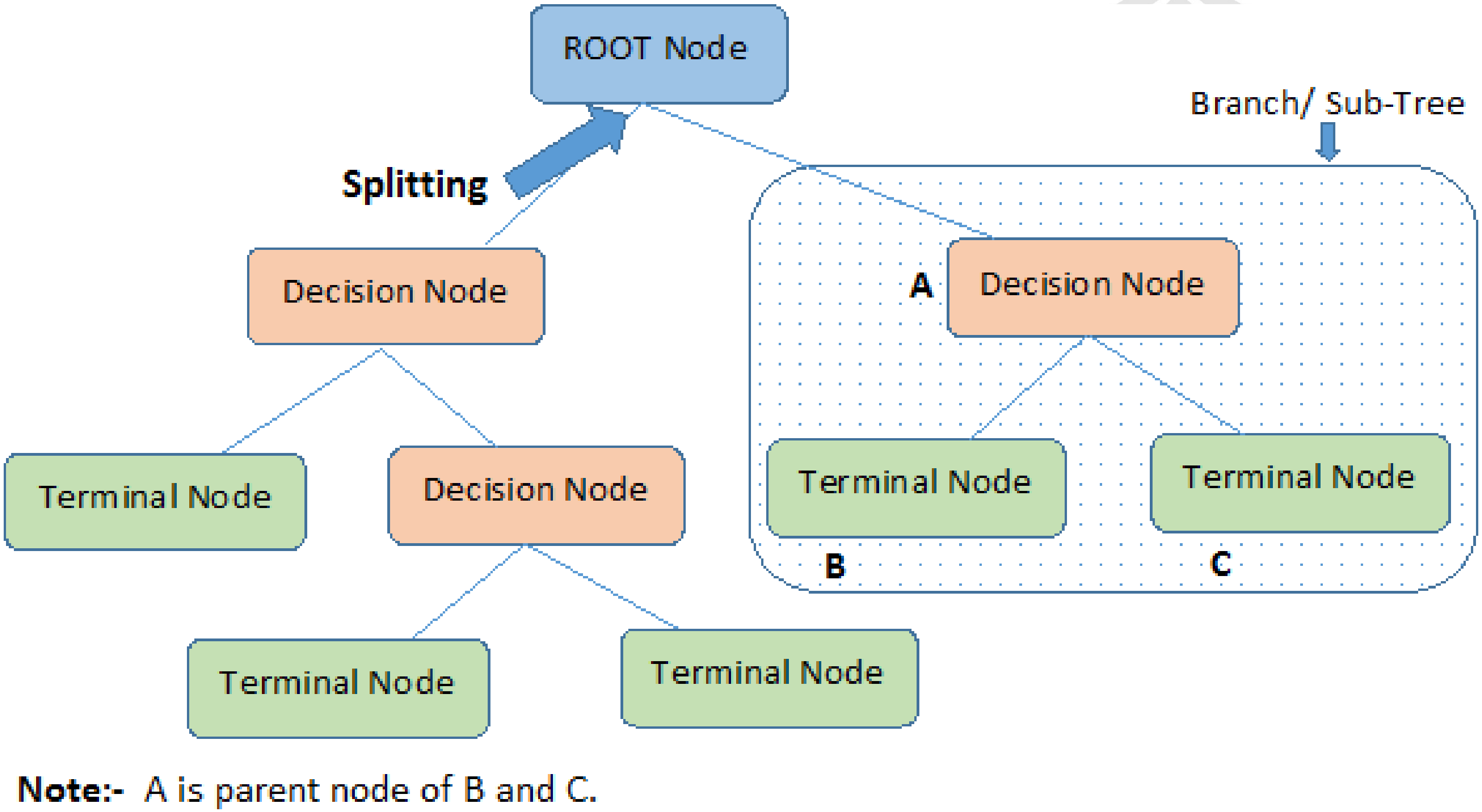
Decision trees in machine learning and data mining

- **Boosted trees** that can be used for regression and classification trees.
- The trees in a **Rotation Forest** are all trained by using PCA on a random portion or the data.

Important Terminology related to decision trees

- **Root node** – it represents entire population of sample and this further gets divided into two or more homogeneous sets.
- **Splitting** – it is a process of dividing a node into two or more sub-nodes.
- **Decision node** – when a sub-node splits into further sub-nodes, then it is called decision node.
- **Leaf / Terminal node** – nodes do not split it called leaf or terminal node.

Important Terminology related to decision trees



Cr: https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

Important Terminology related to decision trees

- **Pruning** – when we remove sub-nodes of a decision node, this process is called pruning.
- **Branch / Sub-tress** – A sub section of entire tree is called branch or sub-tree.
- **Parent and child node** – a node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

Advantages

- **Easy to understand** – it does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.
- **Useful in Data exploration** – decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. It can also be used in data exploration stage. For example, we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable.

Advantages

- **Less data cleaning required** – it requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
- **Data type is not a constraint** – it can handle both numerical and categorical variables
- **Non parametric method** – Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

Disadvantages

- **Over fitting** – over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning.
- **Not fit for continuous variables** – while working with continuous numerical variables, decision tree looses information when it categorizes variables in different categories.

How does a tree decide where to split?

- The decision of making strategic splits heavily affects a tree’s accuracy.
- The decision criteria is different for classification and regression trees.
- Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes.
- The creation of sub-nodes increase the homogeneity of resultant sub-nodes. In other words, we can say that purity of the node increases with respect to the target variable.

How does a tree decide where to split?

- Decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.
- The algorithm selection is also based on type of target variables.

How does a tree decide where to split?

- Let's look at the four most commonly used algorithms in decision tree
  - Gini Index
  - Chi-square
  - Information gain
  - Reduction in variance

Gini Index

- Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure.
  - It works with categorical target variable “Success” or “Failure”.
  - It performs only Binary splits
  - Higher the value of Gini higher the homogeneily.
  - CART (Classification and Regression Tree) uses Gini method to create binary splits.

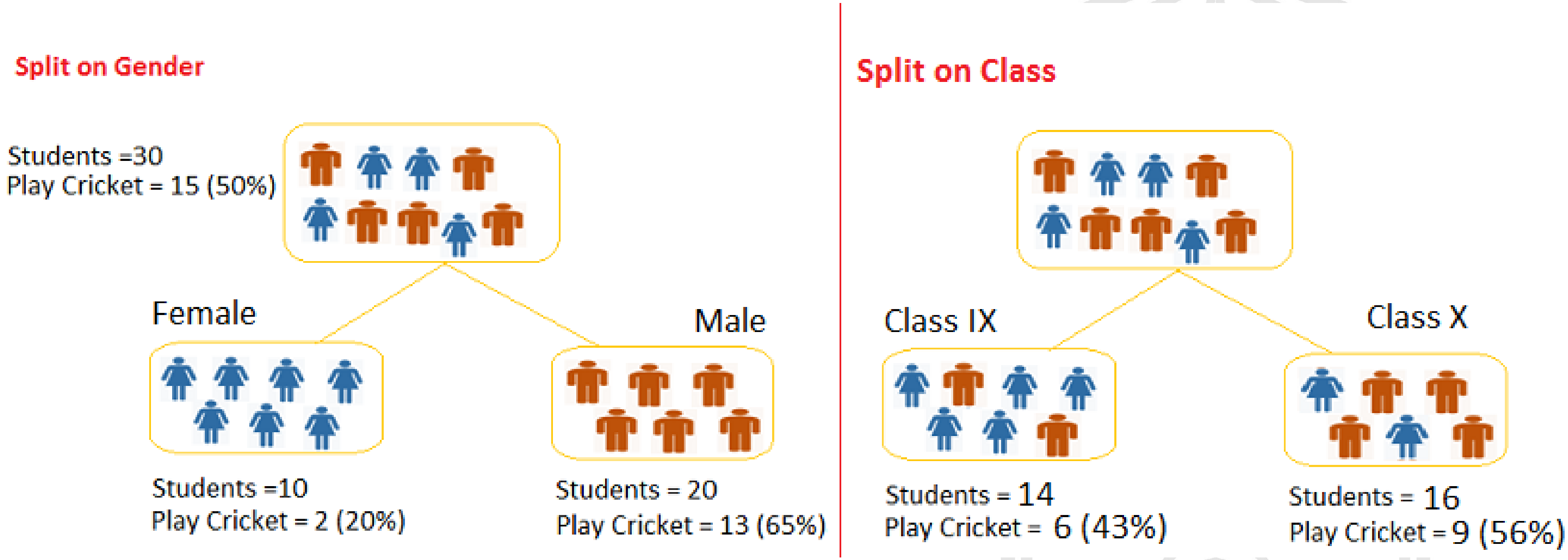
Gini Index

- **Steps to calculate Gini fro a split**
  - Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p^2+q^2)
  - Calculate Gini for split using weighted Gini score of each node of that split

Gini Index

- **Example** – referring to example used above, where we want to segregate the students based on t arget variable (playing cricket or not).
- In the picture below, we split the population using two input variables *Gender* and *Class*.
- Now, I want to identify which split is producing more homogeneous sub-nodes using Gini index.

Gini Index



Split on Gender

Students =30
Play Cricket = 15 (50%)

Female

Students =10
Play Cricket = 2 (20%)

Male

Students = 20
Play Cricket = 13 (65%)

Split on Class

Class IX

Students = 14
Play Cricket = 6 (43%)

Class X

Students = 16
Play Cricket = 9 (56%)

Gini Index

- **Split on Gender:**
  - Calculate, Gini for sub-node Female = (0.2*0.2) + (0.8*0.8) = 0.68
  - Gini for sub-node Male = (0.65*0.65)+(0.35*0.35) = 0.55
  - Calculate weighted Gini for split gender = (10/30)*0.68+(20/30)*0.55 = **0.59**

Gini Index

- **Split on class:**
  - Gini for sub-node Class IX = (0.43*0.43) + (0.57*0.57) = 0.51
  - Gini for sub-node Class X = (0.56*0.56)+(0.44*0.44) = 0.51
  - Calculate weighted Gini for split class = (14/30)*0.51+(16/30)*0.51 = **0.51**
- Above, you can see that Gini score for split on Gender is higher than split on class, hence, the node split will take place on **Gender**.

Chi-square

- It is an algorithm to find out the statistical significance between the differences between sub-nodes and parent node.
- We measure it by sum of squares of standardized differences between observed and expected frequencies of target variable.

Chi-square

- It works with categorical target variable “success” or “failure”.
- It can perform two or more splits.
- Higher the value of Chi-square higher the statistical significance of differences between sub-node and parent node.
- Chi-square of each node is calculated using formula
  - Chi-square = ((actual – expected)^2 / expected)^1/2
- It generates tree called CHAID (Chi-square automatic interaction detector)

Chi-square

- Calculate Chi-square for individual node by calculating the deviation for success and failure both.
- Calculated Chi-square for split using sum of all chi-square of success and failure of each node of the split.

Chi-square

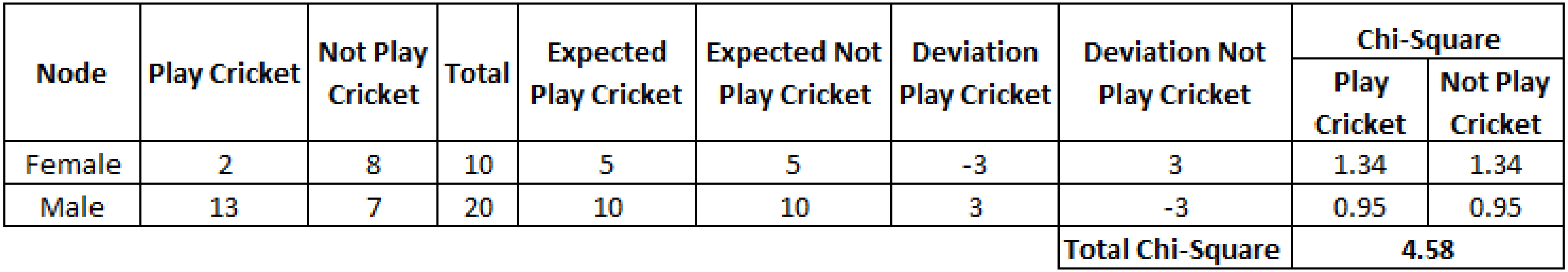
- **Split on Gender:**
  - First we are populating for node Female, populate the actual value for “**Play cricket**” and “**Not play cricket**”, here these are 2 and 8 respectively.
  - Calculate expected value for “**Play cricket**” and “**Not play cricket**”, here it would be 5 for both because parent node has probability of 50% and we have applied same probability of Female (count(10)).

Chi-square

- Calculate deviations by using formula, (Actual – Expected). It is for “**play cricket**” (2-5 = -3) and for “**Not play cricket**” (8-5 = 3).
- Calculate Chi-square of node for “**Play cricket**” and “**Not play cricket**” using formula with formula
  - ((Actual – Expected)^2 / Expected)^1/2

Chi-square

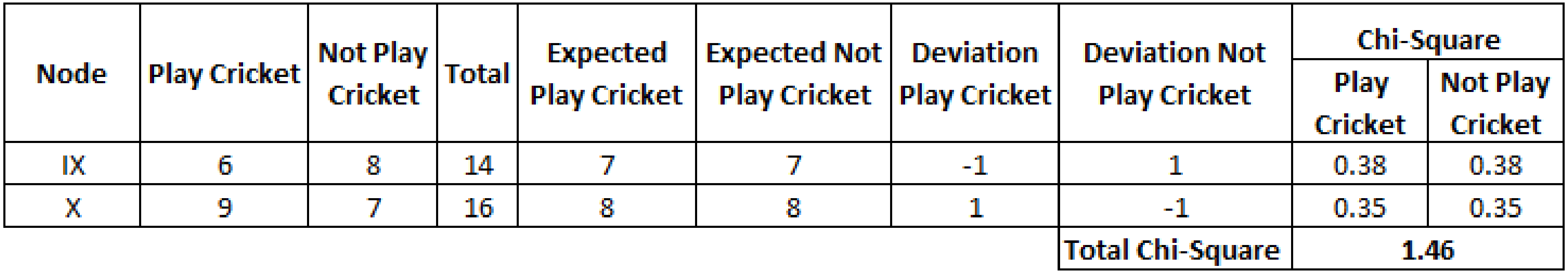
- Follow similar steps for calculating Chi-square value for male node.
- Now add all Chi-square values to calculate Chi-square for split Gender.



Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
Female	2	8	10	5	5	-3	3	1.34	1.34
Male	13	7	20	10	10	3	-3	0.95	0.95
							Total Chi-Square	4.58	

Chi-square

- Perform similar steps of calculation for split on Class and you will come up with below table



Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
IX	6	8	14	7	7	-1	1	0.38	0.38
X	9	7	16	8	8	1	-1	0.35	0.35
							Total Chi-Square	1.46	

- Above, you can see that Chi-square also identify the Gender split is more significant compare to Class.

Reduction in Variance

- Reduction in variance is an algorithm used for continuous target variables (regression problems).
- This algorithm uses the standard formula of variance to choose the best split.
- The split with lower variance is selected as the criteria to split the population

$$\text{Variance} = \frac{\Sigma(X - \overline{X})^2}{n}$$

X-bar is mean of the values,
X is actual
n is number of values

Reduction in Variance

- **Steps to calculate variance**
  - Calculate variance for each node
  - Calculate variance for each split as weighted average of each node variance

Reduction in Variance

- **Example** – let's assign numerical value **1** for play cricket and **0** for not playing cricket.
    - Variance for Root node, here mean value is (15*1 + 15*0)/30 = 0.5 and we have 15 one and 15 zero.
        - Now variance would be ((1-0.5)^2+(1-0.5)^2+…15 times + (0-0.5)^2 + … 15 times) / 30, this can be written as ***(15*(1-0.5)^2+15*(0-0.5)^2) / 30 = 0.25***

Reduction in Variance

- Mean of Female node = (2*1+8*0)/10=0.2 and Variance = (2*(1-0.2)^2+8*(0-0.2)^2) / 10 = 0.16
- Mean of Male Node = (13*1+7*0)/20=0.65 and Variance = (13*(1-0.65)^2+7*(0-0.65)^2) / 20 = 0.23
- Variance for Split Gender = Weighted Variance of Sub-nodes = (10/30)*0.16 + (20/30) *0.23 = **0.21**

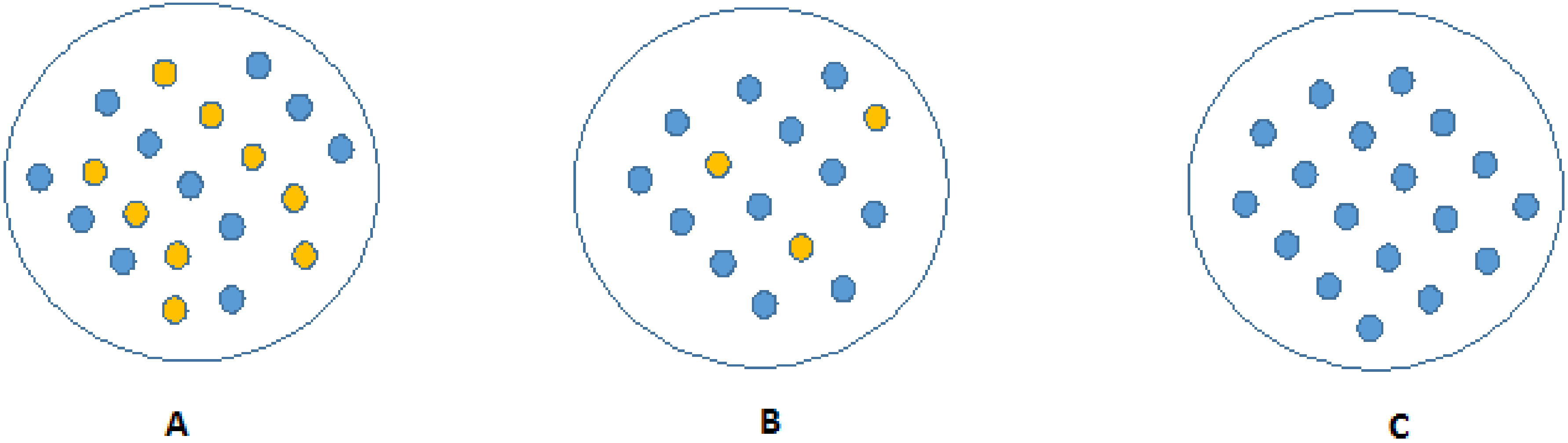
Reduction in Variance

- Mean of Class IX node = (6*1+8*0)/14=0.43 and Variance = (6*(1-0.43)^2+8*(0-0.43)^2) / 14= 0.24
- Mean of Class X node = (9*1+7*0)/16=0.56 and Variance = (9*(1-0.56)^2+7*(0-0.56)^2) / 16 = 0.25
- Variance for Split Gender = (14/30)*0.24 + (16/30) *0.25 = **0.25**

- Above, you can see that **Gender** split has lower variance compare to parent node, so the split would take place on Gender variable.

Information Gain

- Look at the image below and think which node can be described easily. Your answer is C because it requires less information as all values are



Information Gain

- On the other hand, B requires more information to describe it and A requires the maximum information.
- In other words, we can say that C is a Pure node. B is less Impure and A is more impure.

Information Gain

- We can build a conclusion that less impure node requires less information to describe it.
- And, more impure node requires more information.
- Information theory is a measure to define this degree of disorganization in a system known as **Entropy**.
- If the sample is complete homogeneous, then the entropy is **zero** and if the sample is and equally divided (50%-50%), it has entropy of one.

Information Gain

- Entropy can be calculated using formula
-
$$\text{Entropy} = -p\log_2 p - q\log_2 q$$
-
- Here **p** and **q** is probability of success and failure respectively in that node.
- Entropy is also used with categorical target variable.
- It chooses the split which has lowest entropy compared to parent node and other splits.
- The lesser the entropy, the better it is.

- On the other hand, B requires more information to describe it and A requires the maximum information.
- In other words, we can say that C is a Pure node. B is less Impure and A is more impure.

References

- https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/
- http://scikit-learn.org/stable/modules/tree.html
- https://www.lucidchart.com/pages/decision-tree
- http://dataminingtrend.com/2014/decision-tree-model/
- http://dataminingtrend.com/2014/decision-tree-numerical-attributes/
- http://dataminingtrend.com/2014/data-mini

References

- https://hbr.org/1964/07/decision-trees-for-decision-making
- http://heller.brandeis.edu/executive-education/pdfs/DecisionTrees.pdf
-