Lecture 3:Decision Tree

Concepts: Recap

- **Concepts in data mining/statistics**:
  - Data
  - Instance/record/example/data point/observation/row
  - Variable/feature/attribute/column
  - Variable type
    - Categorical variable
      - **Nominal** variables are variables that have two or more categories, but which do not have an intrinsic order.
      - **Ordinal** variables are variables that have two or more categories just like nominal variables only the categories can also be ordered or ranked.
    - Numeric/continuous variable

Concepts: Recap

- **Concepts in R programming**:
  - Data type
    - Vector: stores an ordered set of values called elements.
    - Factor: A factor is a special case of vector that is solely used to represent categorical variables.
    - Dataframe: a structure analogous to a spreadsheet or database, since it has both rows and columns of data.
  - In R, we use Dataframe to store the imported data. Each column of the Dataframe is a vector (or factor).
    - Vector for numeric variables
    - Factor for categorical variables

Concepts: Recap

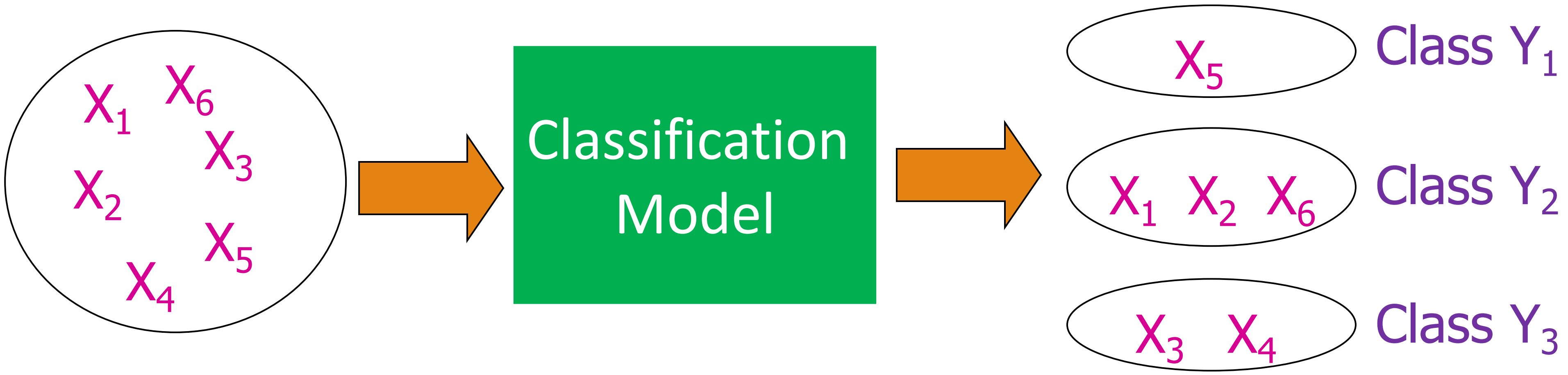
Auction	Color	IsBadBuy	MMRCurrent	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	WHITE	No	2871	LARGE TRUC	FORD	5300	8	75419	869	Alloy
ADESA	GOLD	Yes	1840	VAN	FORD	3600	8	82944	2322	Alloy
ADESA	RED	No	8931	SMALL SUV	CHRYSLER	7500	4	57338	588	Alloy
ADESA	GOLD	No	8320	CROSSOVER	FORD	8500	5	55909	1169	Alloy
ADESA	GREY	No	11520	LARGE TRUC	FORD	10100	5	86702	853	Alloy
ADESA	SILVER	No	2659	COMPACT	GM	4100	7	73810	1455	Covers
ADESA	RED	No	4645	VAN	FORD	5600	5	85003	1633	Covers
ADESA	SILVER	No	4352	LARGE	GM	5900	5	88991	2152	Covers
ADESA	SILVER	No	5142	MEDIUM	GM	6600	5	80077	1373	Alloy
ADESA	MAROON	No	9983	MEDIUM	OTHER	7500	3	71952	1272	Alloy
ADESA	WHITE	No	4165	MEDIUM	OTHER	6200	4	23881	462	Covers

Concepts: Recap

```
'data.frame':   10000 obs. of  11 variables:
 $ Auction                     : Factor w/ 3 levels "ADESA","MANHEIM",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Color                       : Factor w/ 16 levels "BEIGE","BLACK",..: 15 5 13 5 7 14 13 14 14 8 ...
 $ IsBadBuy                    : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...
 $ MMRCurrentAuctionAveragePrice: int  2871 1840 8931 8320 11520 2659 4645 4352 5142 9983 ...
 $ Size                        : Factor w/ 12 levels "COMPACT","CROSSOVER",..: 5 12 8 2 5 1 12 3 6 6 ...
 $ TopThreeAmericanName        : Factor w/ 4 levels "CHRYSLER","FORD",..: 2 2 1 2 2 3 2 3 3 4 ...
 $ VehBCost                    : int  5300 3600 7500 8500 10100 4100 5600 5900 6600 7500 ...
 $ VehicleAge                  : int  8 8 4 5 5 7 5 5 5 3 ...
 $ VehOdo                      : int  75419 82944 57338 55909 86702 73810 85003 88991 80077 71952 ...
 $ WarrantyCost                : int  869 2322 588 1169 853 1455 1633 2152 1373 1272 ...
 $ WheelType                   : Factor w/ 4 levels "Alloy","Covers",..: 1 1 1 1 1 2 2 2 1 1 ...
```

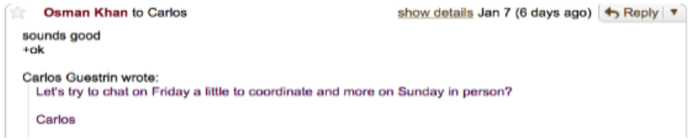
Classification: Recap

- Classify objects into a set of **pre-specified** classes (or categories) based on the values of relevant object attributes (features).



Classes Y_1, Y_2, and Y_3 are pre-determined

Classification: Recap

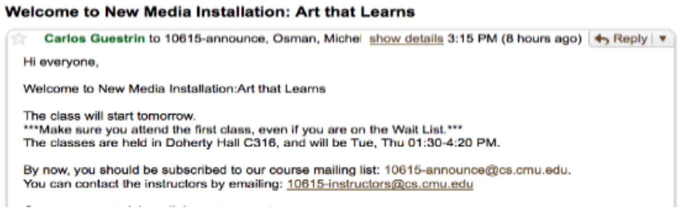


Osman Khan to Carlos show details Jan 7 (6 days ago) Reply

sounds good
+ok

Carlos Guestrin wrote:
Let's try to chat on Friday a little to coordinate and more on Sunday in person?

Carlos



Welcome to New Media Installation: Art that Learns

Carlos Guestrin to 10615-announce, Osman, Michel show details 3:15 PM (8 hours ago) Reply

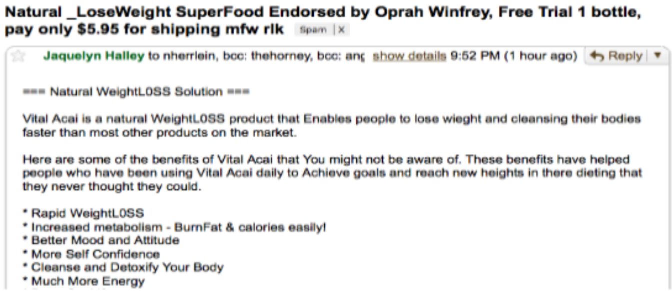
Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Not Spam



Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only $5.95 for shipping mfw rlk Spam | X

Jaquelyn Halley to nherrlein, bcc: thehorney, bcc: ang show details 9:52 PM (1 hour ago) Reply

=== Natural WeightL0SS Solution ===

Vital Acai is a natural WeightL0SS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

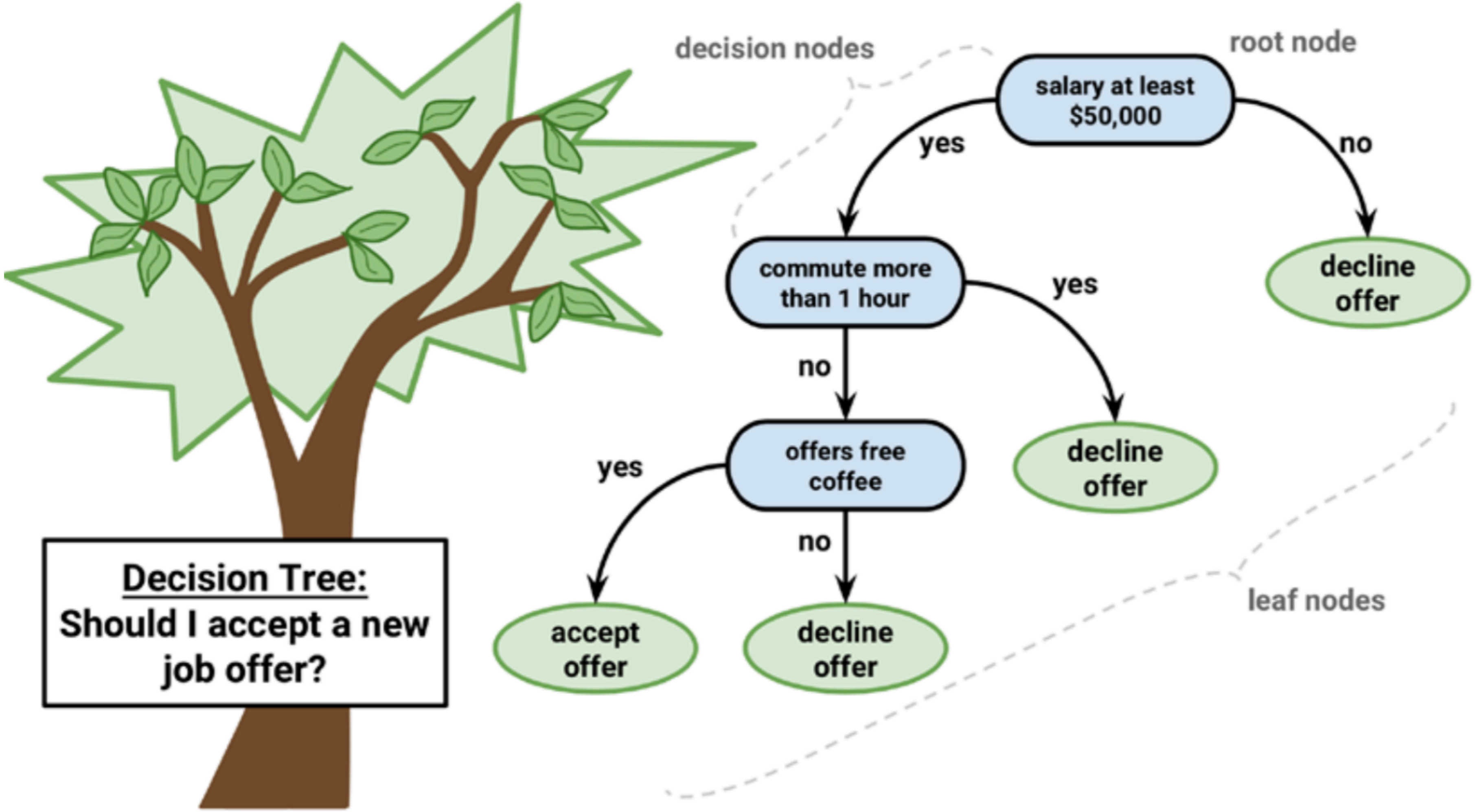
* Rapid WeightL0SS
* Increased metabolism - BurnFat & calories easily!
* Better Mood and Attitude
* More Self Confidence
* Cleanse and Detoxify Your Body
* Much More Energy

Spam

Input: x
Text of email, sender, IP, …

Output: y

Classification: Recap



Overview

- Decision Tree Model
  - Decision Tree Structure
  - Build a Decision Tree
  - Predict with Decision Tree
  - Evaluate Decision Tree Model Performance

Kicked Vehicle

- One of the biggest challenges of an auto dealership purchasing a used car at an auto auction is the risk of that the vehicle might have serious issues that prevent it from being sold to customers. The auto community calls these unfortunate purchases "kicks".
- Kicked cars often result when there are tampered odometers, mechanical issues the dealer is not able to address, issues with getting the vehicle title from the seller, or some other unforeseen problem. Kick cars can be very costly to dealers after transportation cost, throw-away repair work, and market losses in reselling the vehicle.

Kicked Vehicle

- Figure out which cars have a higher risk of being kick can provide real value to dealerships trying to provide the best inventory selection possible to their customers.

- Predict if the car purchased at the Auction is a Kick (bad buy).

What we know about a car?

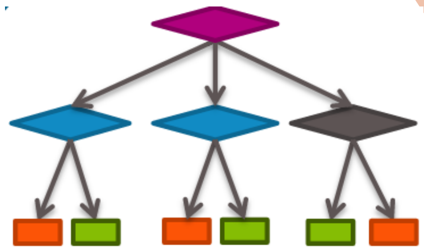
- Auction: Auction provider at which the vehicle was purchased
- Color: Vehicle Color
- MMRCurrentAuctionAveragePrice: Acquisition price for this vehicle in average condition as of current day
- Size: The size category of the vehicle (Compact, SUV, etc.)
- TopThreeAmericanName: Identifies if the manufacturer is one of the top three American manufacturers
- VehBCost: Acquisition cost paid for the vehicle at time of purchase
- VehicleAge: The Years elapsed since the manufacturer's year
- VehOdo: The vehicles odometer reading
- WarrantyCost: Warranty price (term=36month and millage=36K)
- WheelType: The vehicle wheel type description (Alloy, Covers)

- What we don’t know about a car?

What we know about a car?

Information of Cars

- Auction
- Color
- MMRCurrentAuctionAveragePrice
- Size
- TopThreeAmericanName
- VehBCost
- VehicleAge
- VehOdo
- WarrantyCost
- WheelType



IsBadBuy

Predictors (X)

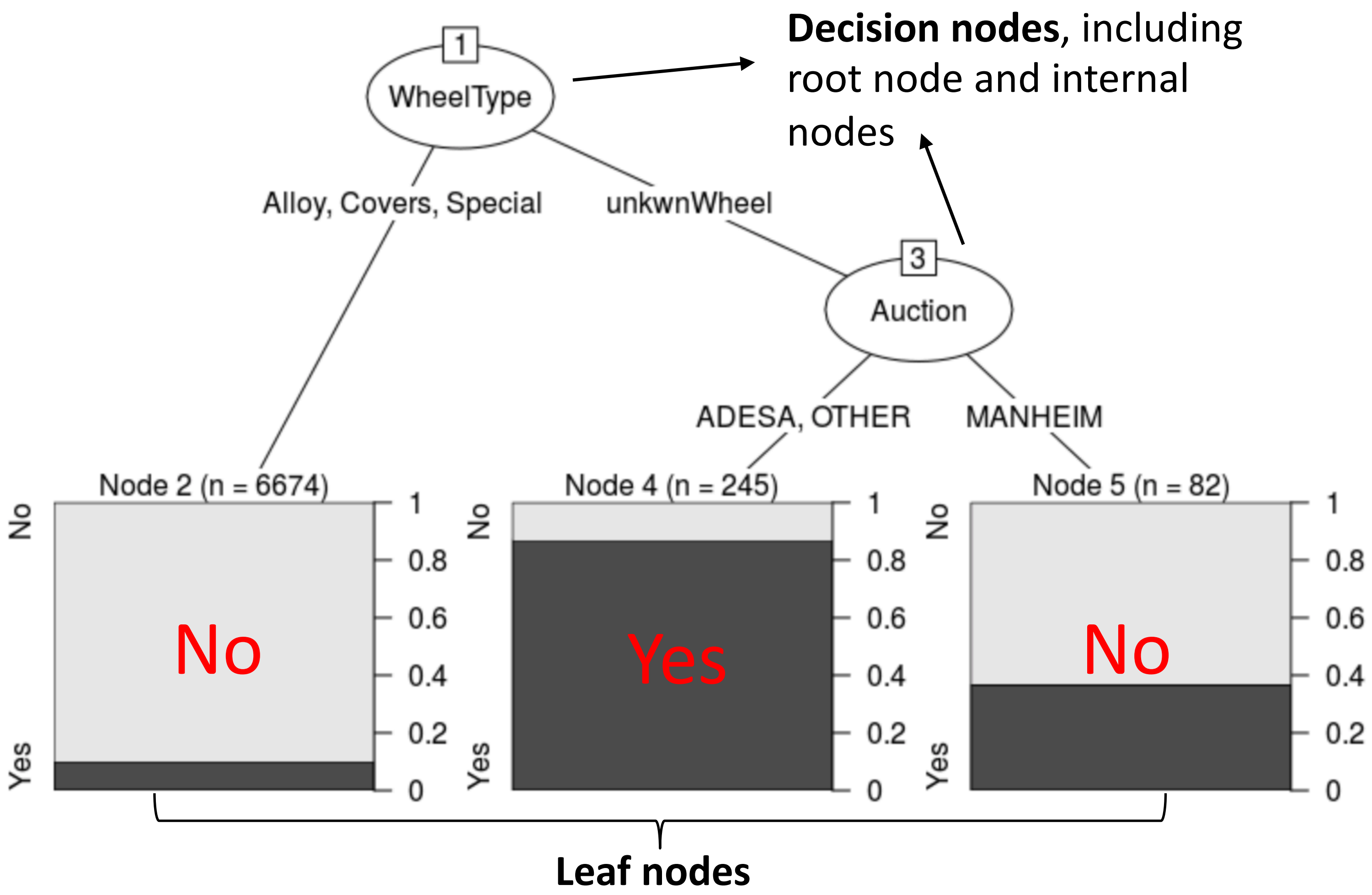
Classifier

Target variable /Label (Y)

Predictor and Target Variable

- **Predictor**: A predictor (or **predictor variable**) is a variable whose values will be used to predict the value of the target variable.
- **Target variable**: The target variable (or **outcome variable**) is the variable whose values are to be modeled and predicted by other variables.
- **Classifier**: an algorithm/classification model that implements classification (mapping input data to a category).

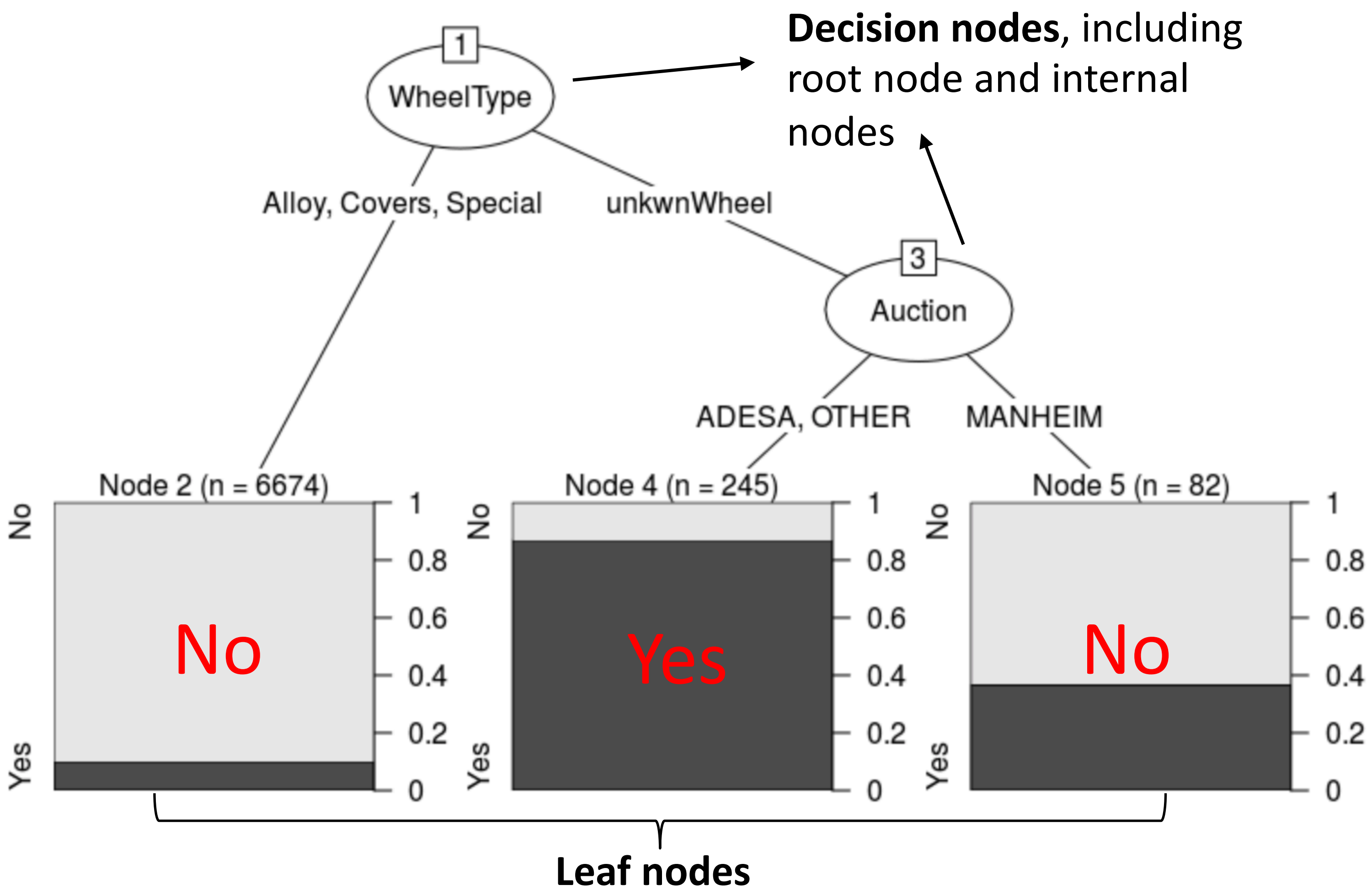
Decision Tree



Decision Tree structure

- **Root Node**: The first node of the tree. It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- **Leaf Node**: The end of the tree. Nodes do not split is called Leaf or Terminal node.
- The root node or an internal node contains a **predictor.**
- **Branches** show feature/attribute values or value ranges
- A leaf node holds a **class label** (**outcome**): prediction result - Yes or No

Decision Tree



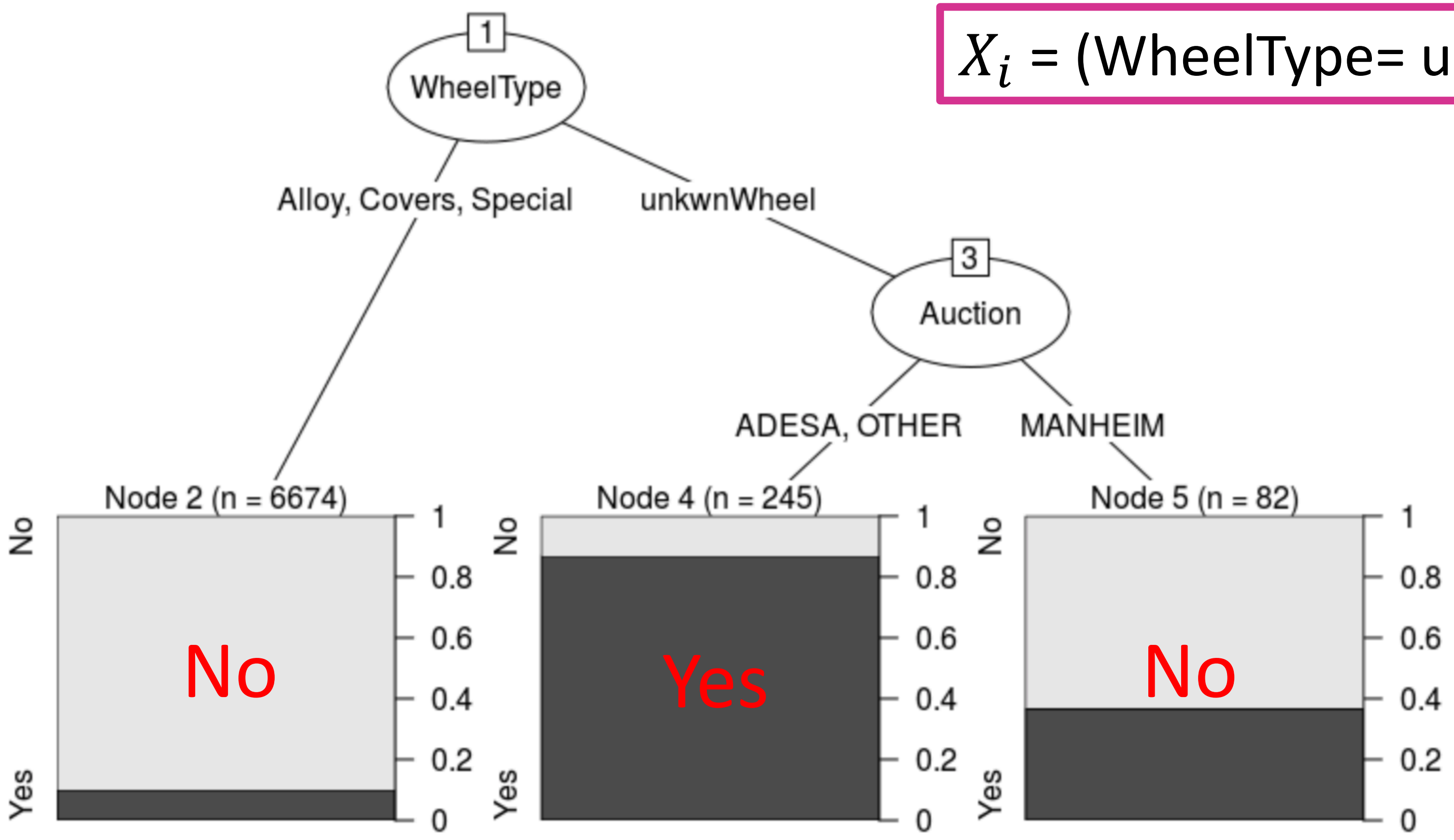
Classification rules

- A decision tree can be expressed as a set of **IF-THEN rules**.
- **Each path from the root to a leaf** forms an IF-THEN rule.
- Each observation/data record finds one unique path from the root to a leaf and is classified into this leaf's class.
- Each observation/data record is classified based on an IF-THEN rule

- **IF** WheelType = unkwnWheel, Auction = OTHER, **THEN** IsBadBuy=YES

Decision Tree

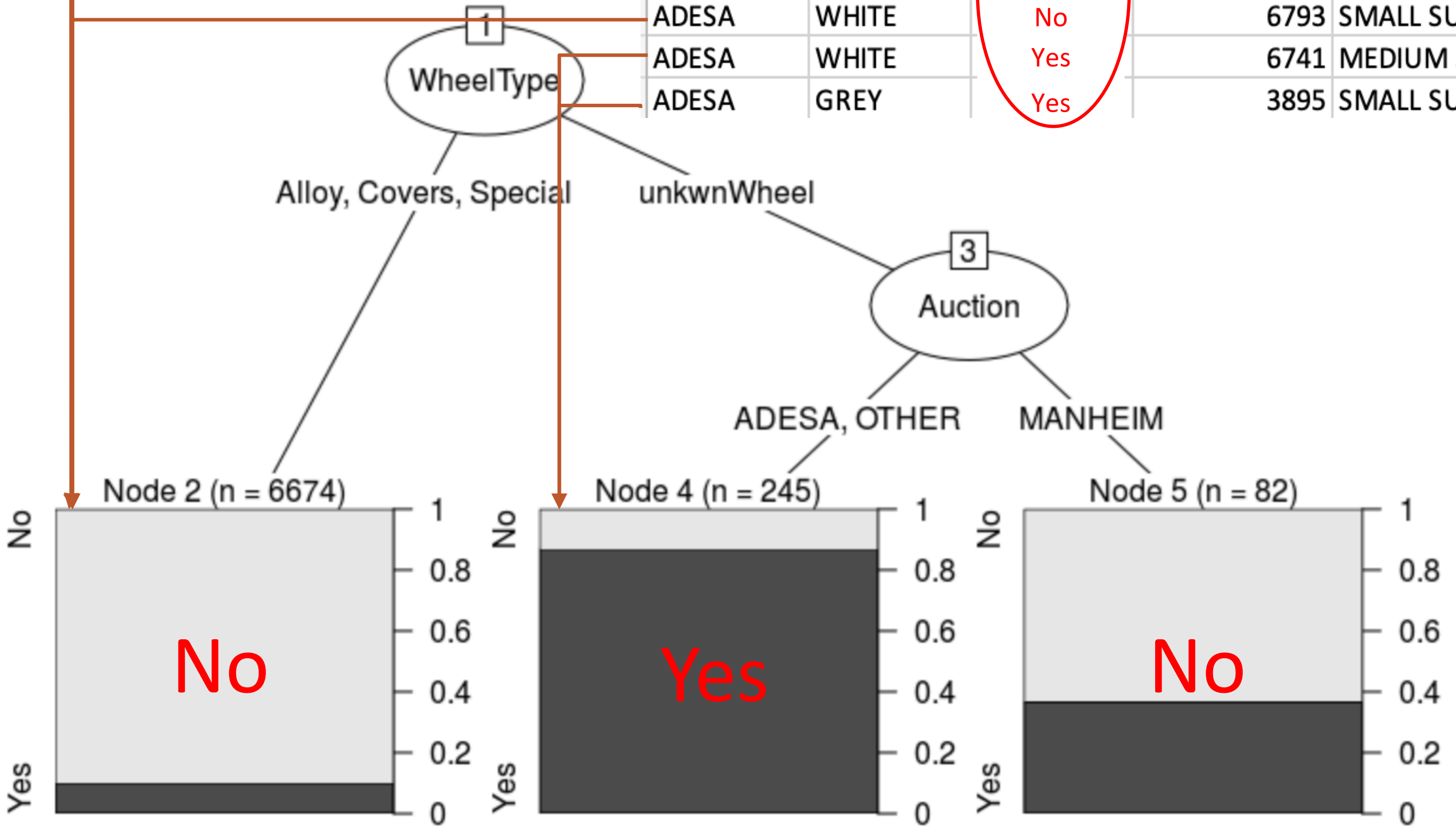
X_i = (WheelType= unkwnWheel, Auction= MANHEIM)



Y_i= No

Decision Tree

Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	RED	No	7860	MEDIUM	CHRYSLER	7500	2	50644	754	Covers
ADESA	SILVER	No	7785	LARGE	GM	8300	3	58384	1500	Alloy
ADESA	BLUE	No	8091	LARGE SUV	FORD	9500	6	80906	1113	Alloy
ADESA	WHITE	No	6793	SMALL SUV	OTHER	7935	5	59801	754	Alloy
ADESA	WHITE	Yes	6741	MEDIUM SU	FORD	9335	6	77178	1740	unkwnWheel
ADESA	GREY	Yes	3895	SMALL SUV	FORD	7100	8	79030	1220	unkwnWheel



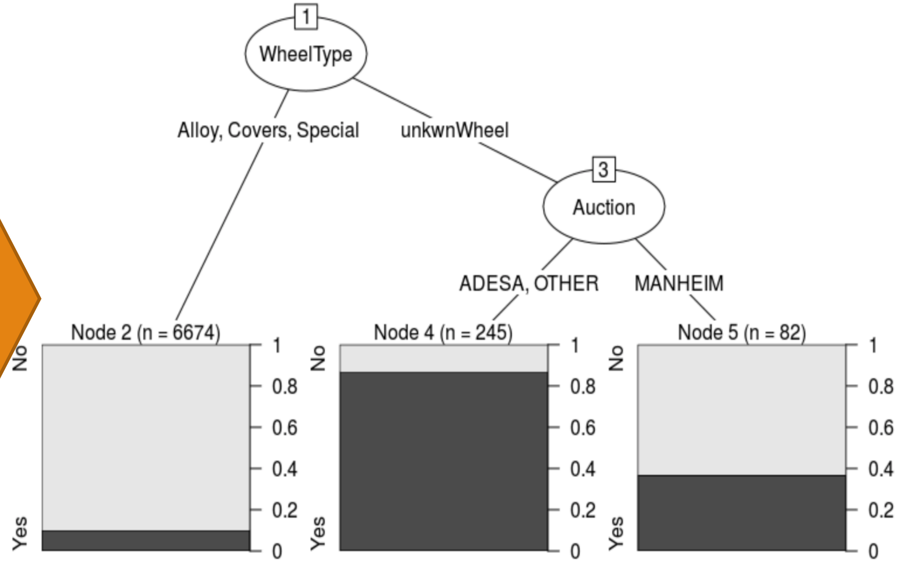
Decision tree splits the dataset

Build a Decision Tree

- Training data: *N* observations $(X_i\ , Y_i\ )$

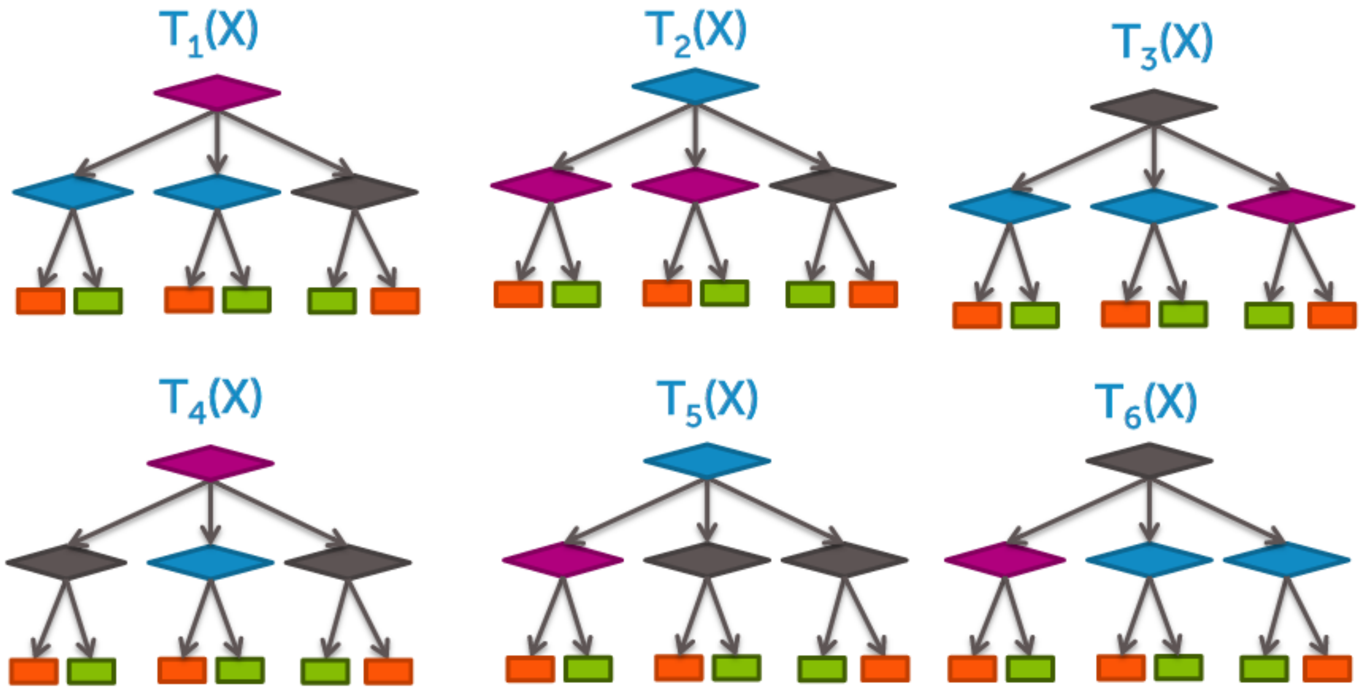
Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	WHITE	No	2871	LARGE TRUC	FORD	5300	8	75419	869	Alloy
ADESA	GOLD	Yes	1840	VAN	FORD	3600	8	82944	2322	Alloy
ADESA	RED	No	8931	SMALL SUV	CHRYSLER	7500	4	57338	588	Alloy
ADESA	GOLD	No	8320	CROSSOVER	FORD	8500	5	55909	1169	Alloy
ADESA	GREY	No	11520	LARGE TRUC	FORD	10100	5	86702	853	Alloy
ADESA	SILVER	No	2659	COMPACT	GM	4100	7	73810	1455	Covers
ADESA	RED	No	4645	VAN	FORD	5600	5	85003	1633	Covers
ADESA	SILVER	No	4352	LARGE	GM	5900	5	88991	2152	Covers
ADESA	SILVER	No	5142	MEDIUM	GM	6600	5	80077	1373	Alloy
ADESA	MAROON	No	9983	MEDIUM	OTHER	7500	3	71952	1272	Alloy
ADESA	WHITE	No	4165	MEDIUM	OTHER	6200	4	23881	462	Covers
ADESA	GOLD	No	2422	VAN	GM	5100	9	83238	5392	Alloy
ADESA	SILVER	No	6603	MEDIUM	OTHER	7300	3	68165	728	Covers
ADESA	GREEN	No	6149	LARGE	FORD	6600	5	93346	1774	Alloy
ADESA	SILVER	Yes	6057	MEDIUM	CHRYSLER	6400	3	73963	1389	Covers
ADESA	SILVER	No	8113	SPECIALTY	CHRYSLER	10400	5	64839	1215	Alloy
ADESA	RED	No	6702	MEDIUM	GM	7100	4	63151	923	Covers
ADESA	MAROON	No	3320	MEDIUM	GM	4700	7	92782	1209	Alloy
ADESA	GREY	No	7708	SPECIALTY	CHRYSLER	9400	5	72592	1389	Alloy
ADESA	WHITE	No	2700	MEDIUM	GM	3900	8	88667	2712	Alloy
ADESA	RED	No	7860	MEDIUM	CHRYSLER	7500	2	50644	754	Covers
ADESA	SILVER	No	7785	LARGE	GM	8300	3	58384	1500	Alloy
ADESA	BLUE	No	8091	LARGE SUV	FORD	9500	6	80906	1113	Alloy
ADESA	WHITE	No	6793	SMALL SUV	OTHER	7935	5	59801	754	Alloy
ADESA	WHITE	No	6741	MEDIUM SU	FORD	9335	6	77178	1740	unkwnWheel
ADESA	GREY	No	3895	SMALL SUV	FORD	7100	8	79030	1220	unkwnWheel
ADESA	SILVER	Yes	6554	MEDIUM	OTHER	6700	4	61315	728	Alloy
ADESA	SILVER	No	2988	MEDIUM	GM	4700	9	92792	2651	Alloy
ADESA	GREY	No	5396	SPORTS	FORD	6600	6	82271	853	Alloy

Minimize classification error on training data



Build a Decision Tree

- Exponentially large number of possible trees makes decision tree learning hard!



Build a Decision Tree

- Greedy Approach to find a "good" tree
  - Step 1: Start with an empty tree
  - Step 2: Select a feature with highest information gain to split data
  - Step 3: Create a branch for each value of the split attribute and according to this, divide the data set into several subsets.
  - Step 4: For each subset:
    - Go to Step 2 & continue (recurse) to split subset

Problem 1: Feature split selection

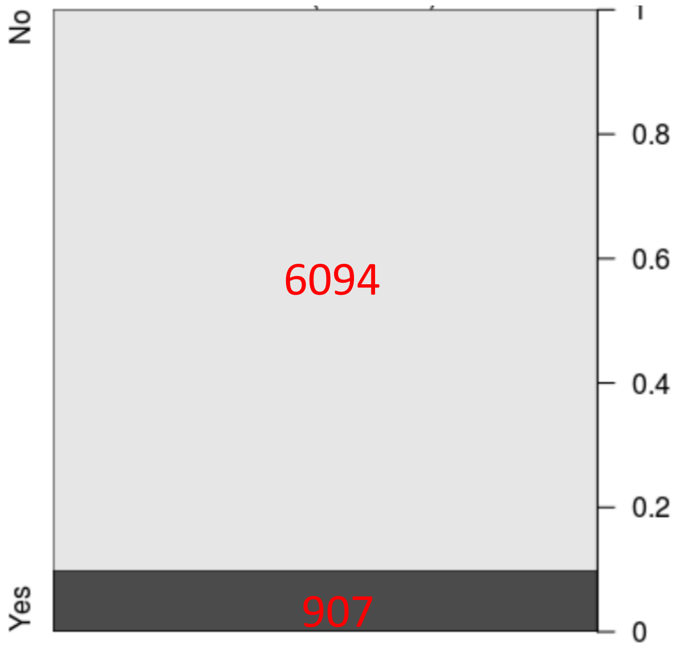
Recursion

Build a Decision Tree

- Key step : Select a feature to split data

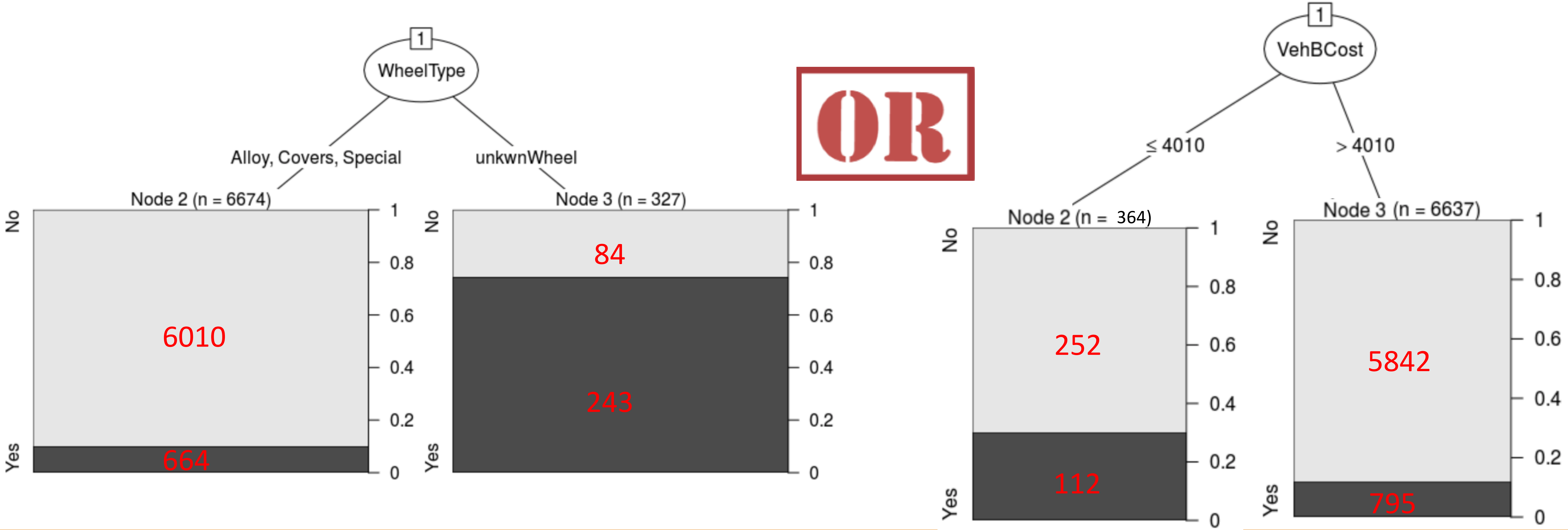
Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	WHITE	No	2871	LARGE TRUC	FORD	5300	8	75419	869	Alloy
ADESA	GOLD	Yes	1840	VAN	FORD	3600	8	82944	2322	Alloy
ADESA	RED	No	8931	SMALL SUV	CHRYSLER	7500	4	57338	588	Alloy
ADESA	GOLD	No	8320	CROSSOVER	FORD	8500	5	55909	1169	Alloy
ADESA	GREY	No	11520	LARGE TRUC	FORD	10100	5	86702	853	Alloy
ADESA	SILVER	No	2659	COMPACT	GM	4100	7	73810	1455	Covers
ADESA	RED	No	4645	VAN	FORD	5600	5	85003	1633	Covers
ADESA	SILVER	No	4352	LARGE	GM	5900	5	88991	2152	Covers
ADESA	SILVER	No	5142	MEDIUM	GM	6600	5	80077	1373	Alloy
ADESA	MAROON	No	9983	MEDIUM	OTHER	7500	3	71952	1272	Alloy
ADESA	WHITE	No	4165	MEDIUM	OTHER	6200	4	23881	462	Covers
ADESA	GOLD	No	2422	VAN	GM	5100	9	83238	5392	Alloy
ADESA	SILVER	No	6603	MEDIUM	OTHER	7300	3	68165	728	Covers
ADESA	GREEN	No	6149	LARGE	FORD	6600	5	93346	1774	Alloy
ADESA	SILVER	Yes	6057	MEDIUM	CHRYSLER	6400	3	73963	1389	Covers
ADESA	SILVER	No	8113	SPECIALTY	CHRYSLER	10400	5	64839	1215	Alloy
ADESA	RED	No	6702	MEDIUM	GM	7100	4	63151	923	Covers
ADESA	MAROON	No	3320	MEDIUM	GM	4700	7	92782	1209	Alloy
ADESA	GREY	No	7708	SPECIALTY	CHRYSLER	9400	5	72592	1389	Alloy
ADESA	WHITE	No	2700	MEDIUM	GM	3900	8	88667	2712	Alloy
ADESA	RED	No	7860	MEDIUM	CHRYSLER	7500	2	50644	754	Covers
ADESA	SILVER	No	7785	LARGE	GM	8300	3	58384	1500	Alloy
ADESA	BLUE	No	8091	LARGE SUV	FORD	9500	6	80906	1113	Alloy
ADESA	WHITE	No	6793	SMALL SUV	OTHER	7935	5	59801	754	Alloy
ADESA	WHITE	No	6741	MEDIUM SU	FORD	9335	6	77178	1740	unkwnWheel
ADESA	GREY	No	3895	SMALL SUV	FORD	7100	8	79030	1220	unkwnWheel
ADESA	SILVER	Yes	6554	MEDIUM	OTHER	6700	4	61315	728	Alloy
ADESA	SILVER	No	2988	MEDIUM	GM	4700	9	92792	2651	Alloy
ADESA	GREY	No	5396	SPORTS	FORD	6600	6	82271	853	Alloy

No split



Build a Decision Tree

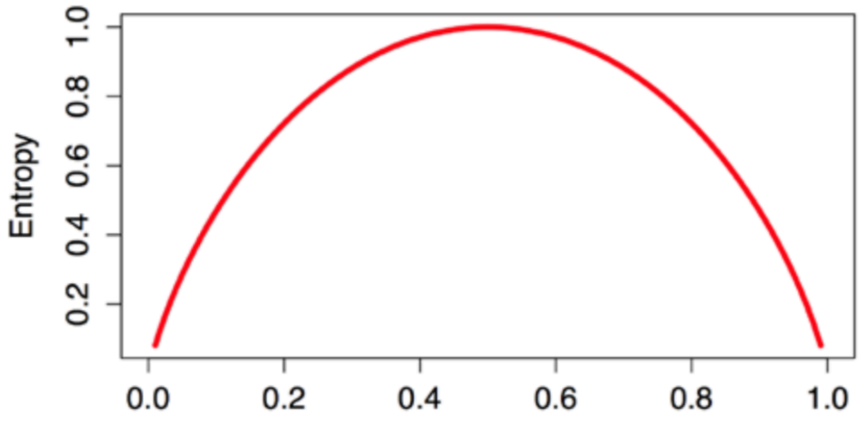
- Key step : Select a feature to split data



Build a Decision Tree

- The degree to which a subset of examples contains only a single class is known as **purity**
- Measure the purity
  - **Entropy**, a concept borrowed from information theory that quantifies the randomness, or disorder, within a set of class values.

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \, log_2(p_i)$$



Build a Decision Tree

- To use entropy to determine the optimal feature to split upon, the algorithm calculates the change in homogeneity, which is a measure known as **information gain**. The information gain for a feature *F* is calculated as the difference between the entropy in the segment before the split *(S1)* and the partitions resulting from the split *(S2)*:

$$\text{InfoGain}(F) = \text{Entropy}(S_1) - \text{Entropy}(S_2)$$

$$\text{Entropy}(S) = \sum_{i=1}^{n} w_i \, \text{Entropy}(P_i)$$

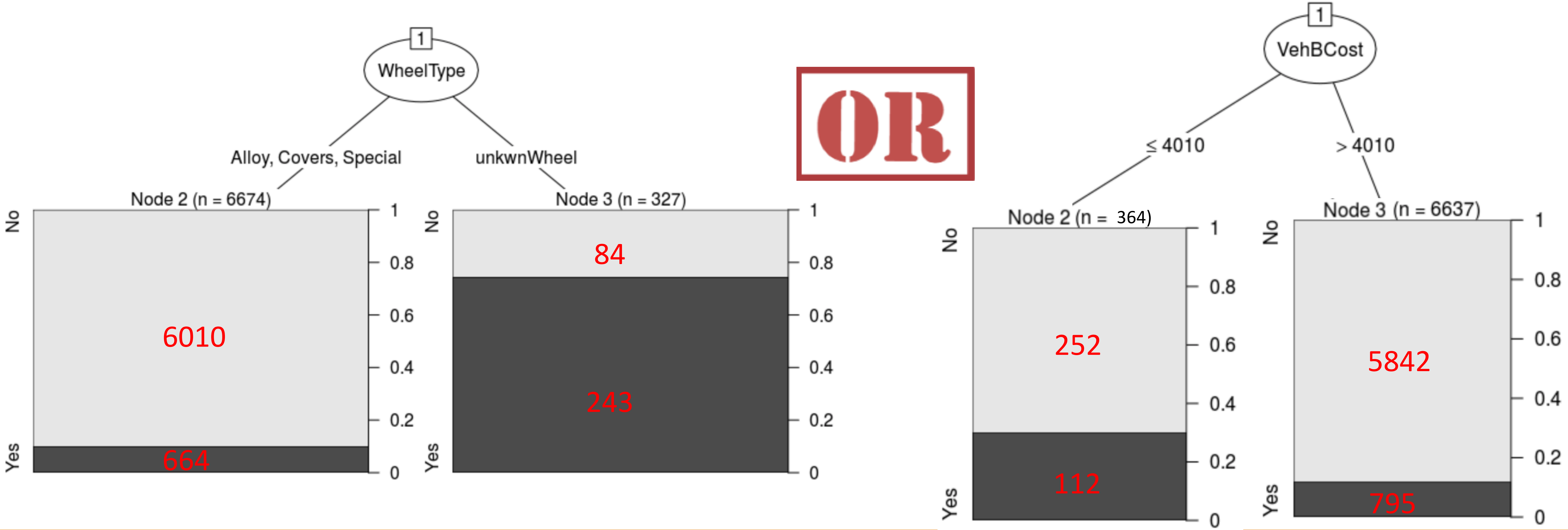
Build a Decision Tree

- The higher the information gain, the better a feature is at creating homogeneous groups after a split on this feature. If the information gain is zero, there is no reduction in entropy for splitting on this feature.

Higher information gain = lower entropy

Build a Decision Tree

- Key step : Select a feature to split data

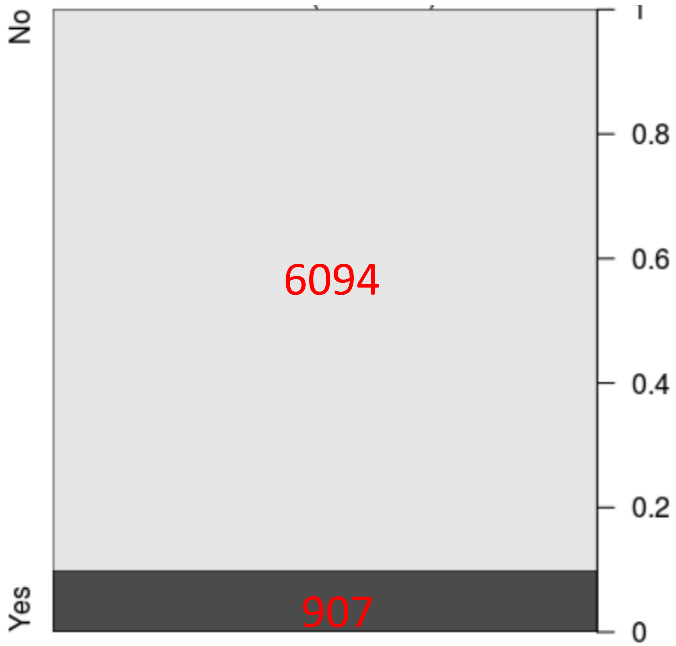


Build a Decision Tree

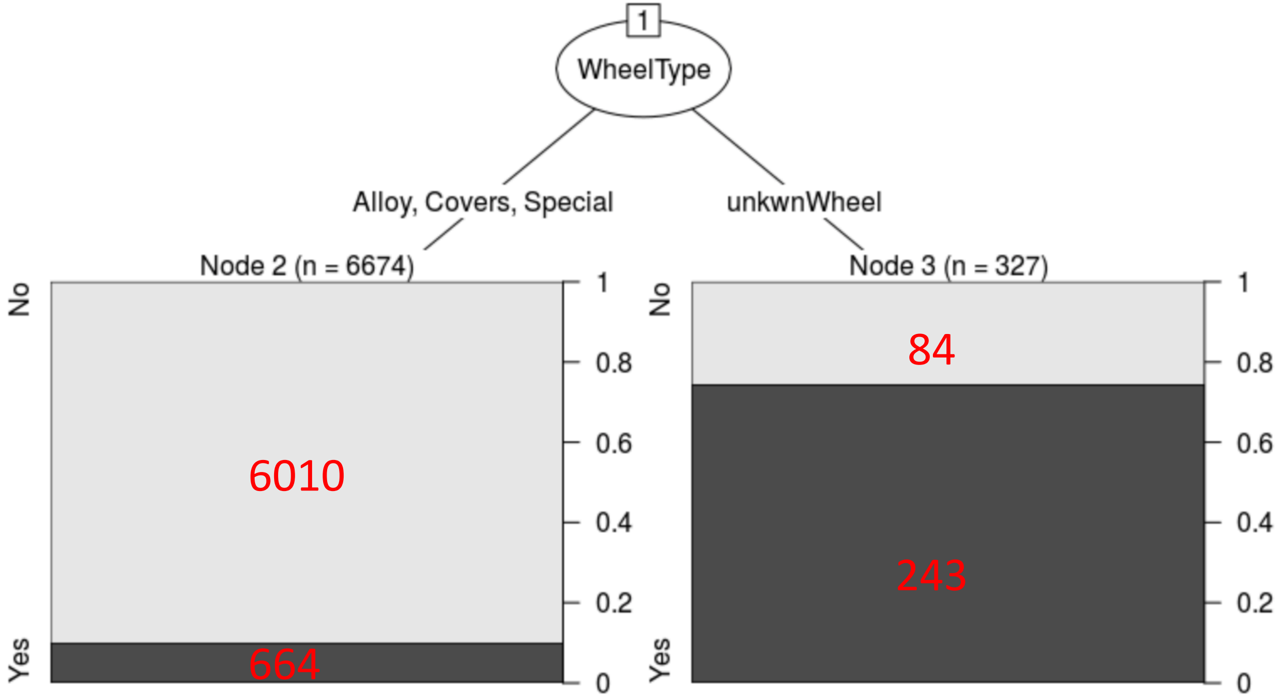
- Entropy before split:
  - Entropy (S_1) =

$$-\frac{6094}{7001}\log_2\frac{6094}{7001} - \frac{907}{7001}\log_2\frac{907}{7001} =$$

$$-0.870 * -0.200 - 0.130 * -2.948 = 0.556$$



Build a Decision Tree



- Entropy after split
  - Entropy (S_2) = $w_1 * Entropy_1 + w_2 * Entropy_2$
    - $w_1 * Entropy_1$= $\frac{6674}{7001}\left(-\frac{6010}{6674}\log_2\frac{6010}{6674}-\frac{664}{6674}\log_2\frac{664}{6674}\right)=0.4455$
    - $w_2 * Entropy_2$= $\frac{327}{7001}\left(-\frac{84}{327}\log_2\frac{84}{327}-\frac{243}{327}\log_2\frac{243}{327}\right)=0.0384$

Entropy (S_2) =0.484

Information gain = 0.556 – 0.484 = 0.072

Build a Decision Tree

- Entropy after split
  - Entropy (S_2) = $w_1 * Entropy_1 + w_2 * Entropy_2$
    - $w_1 * Entropy_1$=

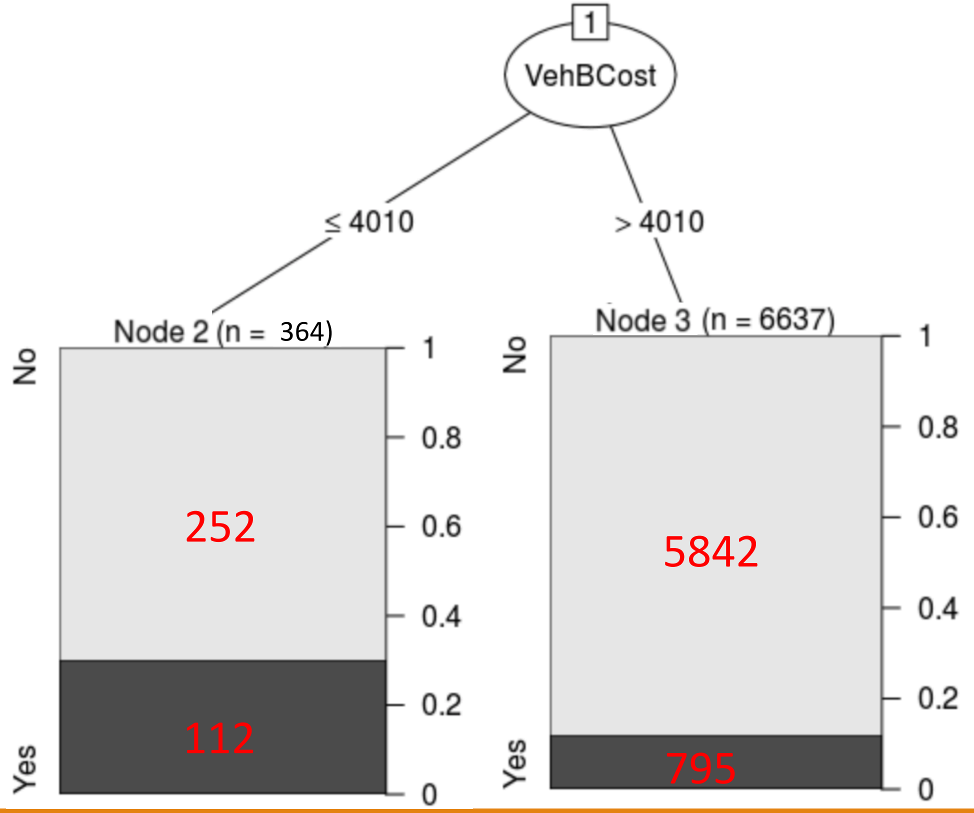
$$\frac{364}{7001}\left(-\frac{252}{364}\log_2\frac{252}{364}-\frac{112}{364}\log_2\frac{112}{364}\right)=0.0463$$

    - $w_2 * Entropy_2$=

$$\frac{6637}{7001}\left(-\frac{5842}{6637}\log_2\frac{5842}{6637}-\frac{795}{6637}\log_2\frac{795}{6637}\right)=0.5012$$

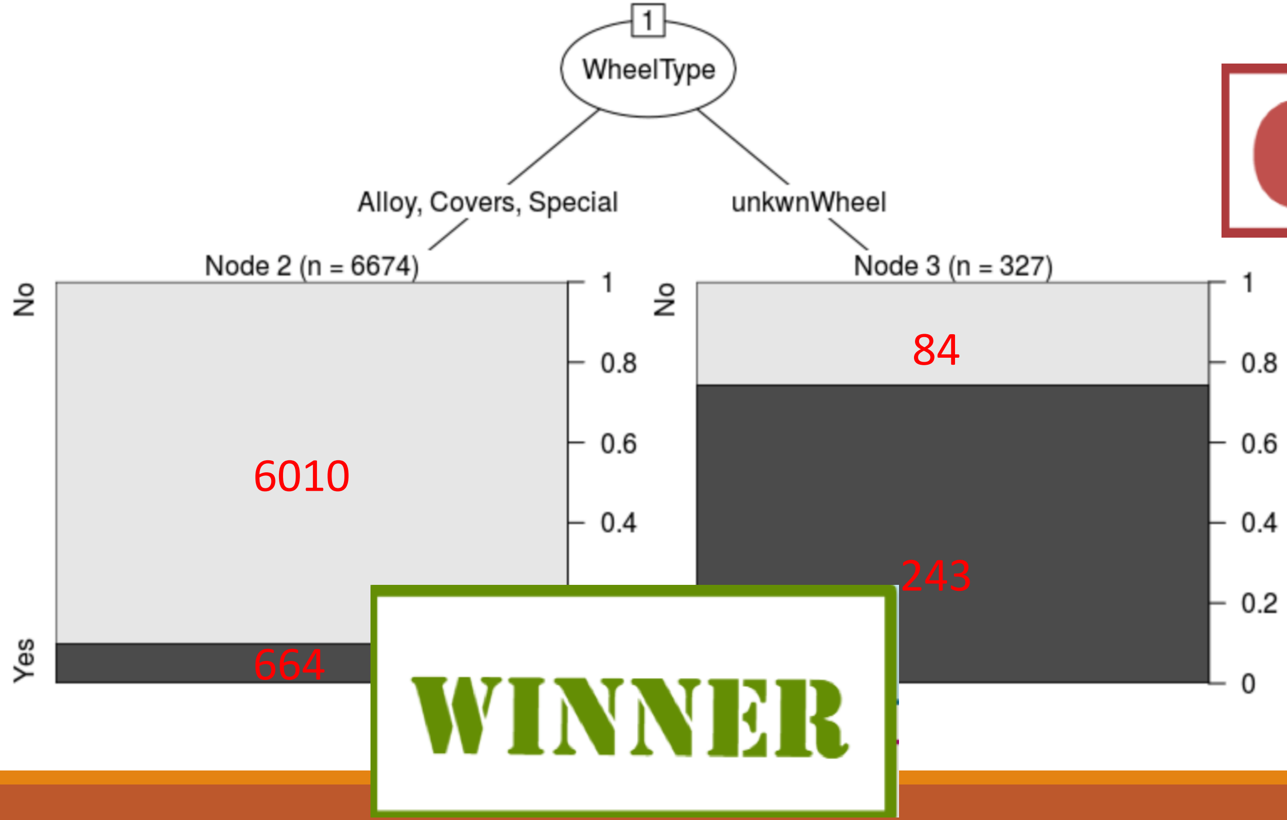
Entropy (S_2) =0.548

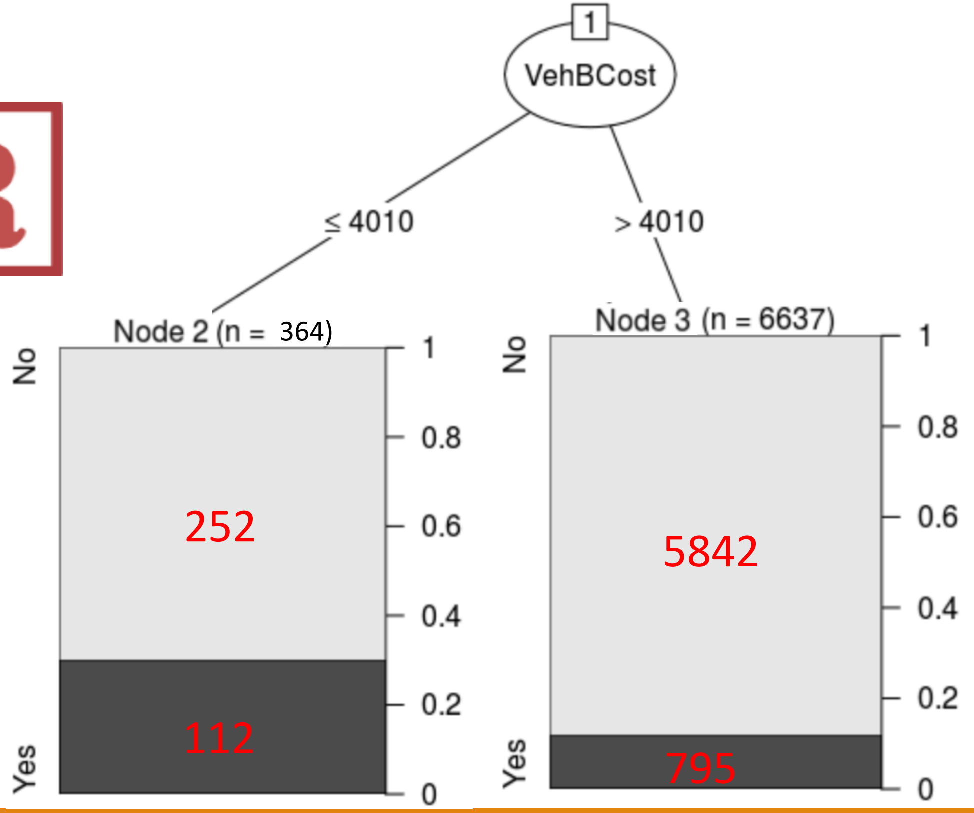
Information gain = 0.556 – 0.548 = 0.008



Build a Decision Tree

- Key step : Select a feature to split data





Build a Decision Tree

- Key step : Select a feature to split data
  - Given a data set D
  - For each feature:
    - Split data of D according to each feature
    - Compute information gain on each split
  - Choose the feature with the highest information gain

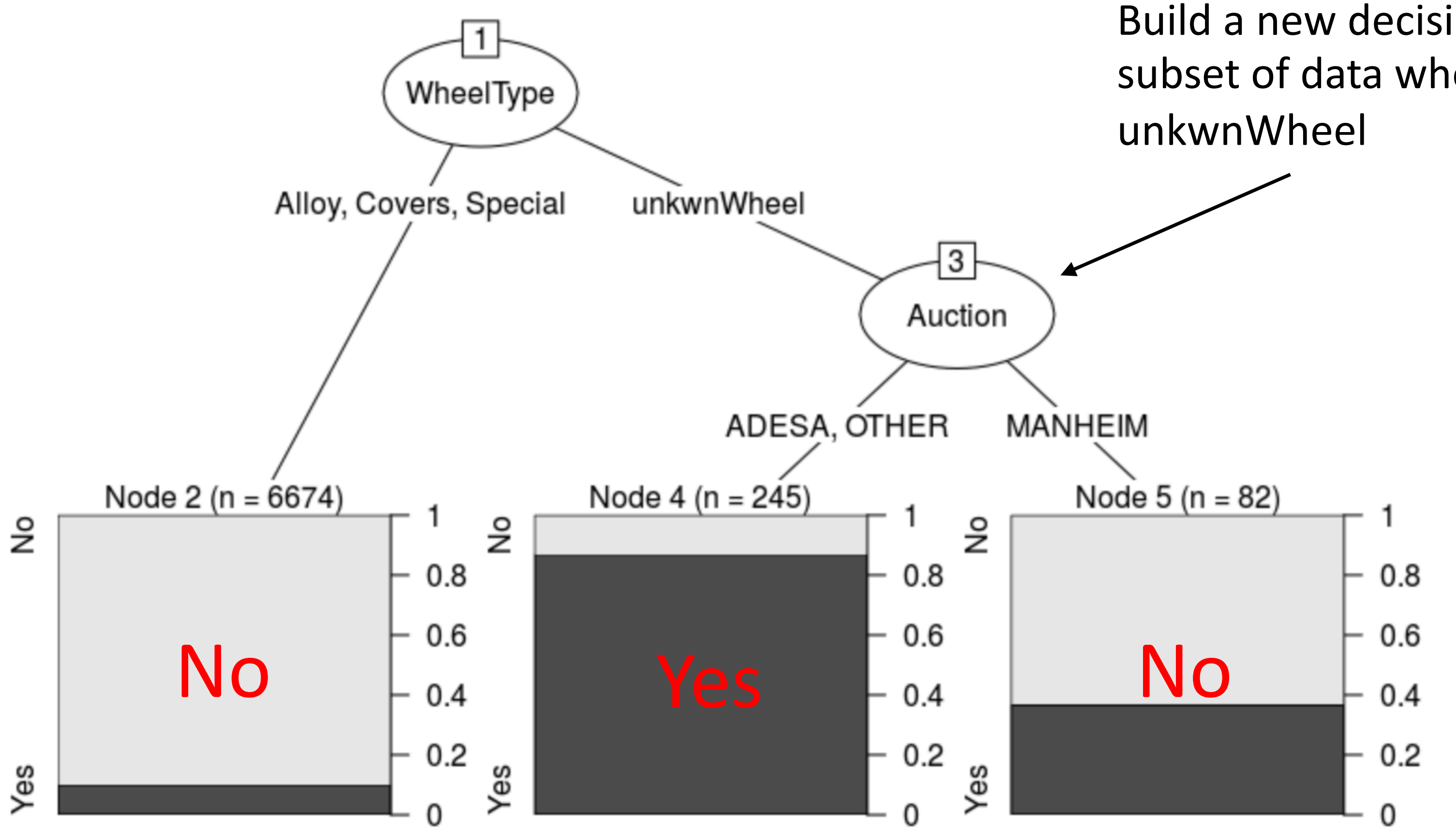
Build a Decision Tree

- Continue to split
  - Build decision node with subset of data where WheelType = Alloy, Cover, or Special
  - Build decision node with subset of data where WheelType = unkwnWheel

Build a Decision Tree

Build a new decision node with subset of data where WheelType = unkwnWheel

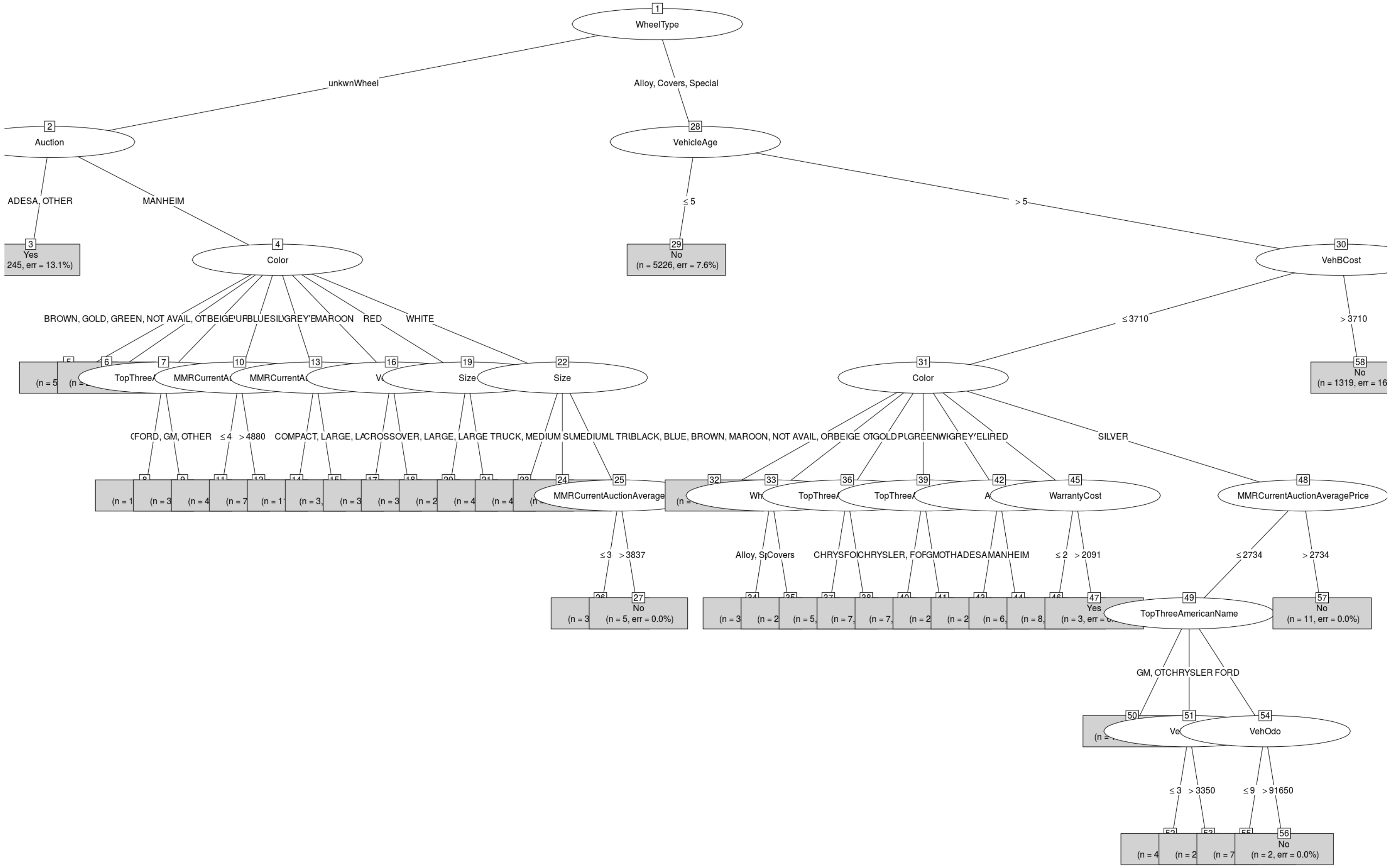
Stop and create a leaf node



Build a Decision Tree

- A decision tree can continue to grow indefinitely, choosing splitting features and dividing the data into smaller and smaller partitions.
  - However, if the tree grows overly large, many of the decisions it makes will be overly specific and the model will be overfitted to the training data.

Build a Decision Tree



Build a Decision Tree

- The process of **pruning** a decision tree involves reducing its size such that it generalizes better to unseen data.
  - **Early stopping/pre-pruning:** stop the tree from growing once it reaches a certain number of decisions
  - **Post-pruning:** growing a tree that is intentionally too large and pruning leaf nodes to reduce the size of the tree to a more appropriate level.

Build a Decision Tree

- Early Stopping conditions:
  - All the records in a node belong to the same class
  - All records in a node have similar attribute values
  - A minimum pre-specified number of records belong to a node

Build a Decision Tree

- Post-pruning: It first grows a large tree that overfits the training data. Later, the nodes and branches that have little effect on the classification errors are removed.

Build a Decision Tree

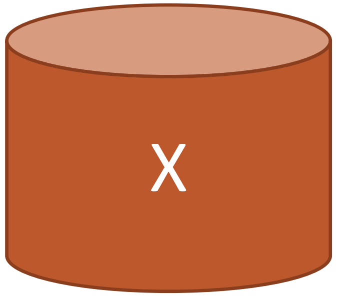
- Greedy Approach to find a "good" tree
  - Step 1: Start with an empty tree
  - Step 2: Select a feature with highest information gain to split data
  - Step 3: Create a branch for each value of the split attribute and according to this, divide the data set into several subsets.
  - Step 4: For each subset:
    - If nothing more to do, create a leaf node
    - Otherwise, go to Step 2 & continue (recurse) to split subset
- Tree pruning (generally, we refers to post-pruning)

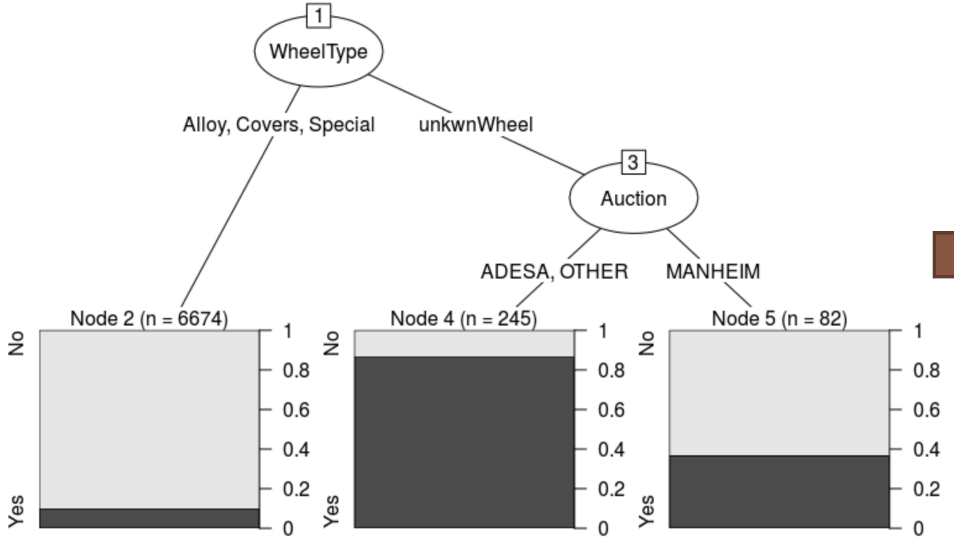
Problem 1: Feature split selection

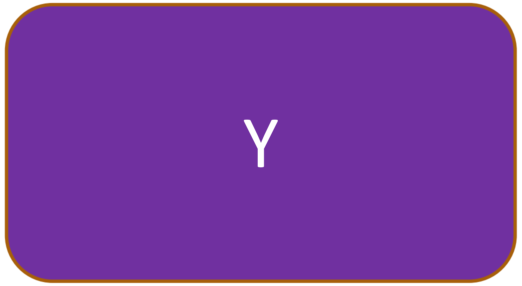
Problem 2: Stopping condition

Recursion

Predictions with Decision Tree







Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	RED		7860	MEDIUM	CHRYSLER	7500	2	50644	754	Covers
ADESA	SILVER		7785	LARGE	GM	8300	3	58384	1500	Alloy
ADESA	BLUE		8091	LARGE SUV	FORD	9500	6	80906	1113	Alloy
ADESA	WHITE		6793	SMALL SUV	OTHER	7935	5	59801	754	Alloy
ADESA	WHITE		6741	MEDIUM SU	FORD	9335	6	77178	1740	unkwnWheel
ADESA	GREY		3895	SMALL SUV	FORD	7100	8	79030	1220	unkwnWheel

IsBadBuy
No
No
No
No
Yes
Yes

Evaluate Decision Tree Model Performance

- Holdout Evaluation
  - Using training data to derive the model and then estimate the accuracy of the learned model can result in over-optimistic estimates due to over-specialization of the model to the data (overfitting).
  - Instead, use holdout testing data that was NOT used to train the model!

Evaluate Decision Tree Model Performance

- How to extract training and holdout testing data sets from one dataset?
  - Approaches
    - Splitting method (percentage split)- divide into training and testing sets (e.g. 70%/30% or 2/3 to 1/3)
    - Random sub-sampling (Random sample pairs)
      - Splitting is repeated n times to generate n different training and hold-out testing pairs.
    - Cross-validation (e.g. 5 or 10 fold)

Evaluate Decision Tree Model Performance

- Splitting method (percentage split)- divide into training and testing sets (e.g. 70%/30% or 2/3 to 1/3)
  - 2 partitions, e.g.
    - 70% and 30% in training and testing
    - Training and testing data are not overlapped
    - Most commonly used to get a sense of a model’s performance level

Evaluate Decision Tree Model Performance

Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	WHITE	No	2871	LARGE TRUC	FORD	5300	8	75419	869	Alloy
ADESA	GOLD	Yes	1840	VAN	FORD	3600	8	82944	2322	Alloy
ADESA	RED	No	8931	SMALL SUV	CHRYSLER	7500	4	57338	588	Alloy
ADESA	GOLD	No	8320	CROSSOVER	FORD	8500	5	55909	1169	Alloy
ADESA	GREY	No	11520	LARGE TRUC	FORD	10100	5	86702	853	Alloy
ADESA	SILVER	No	2659	COMPACT	GM	4100	7	73810	1455	Covers
ADESA	RED	No	4645	VAN	FORD	5600	5	85003	1633	Covers
ADESA	SILVER	No	4352	LARGE	GM	5900	5	88991	2152	Covers
ADESA	SILVER	No	5142	MEDIUM	GM	6600	5	80077	1373	Alloy
ADESA	MAROON	No	9983	MEDIUM	OTHER	7500	3	71952	1272	Alloy
ADESA	WHITE	No	4165	MEDIUM	OTHER	6200	4	23881	462	Covers
ADESA	GOLD	No	2422	VAN	GM	5100	9	83238	5392	Alloy
ADESA	SILVER	No	6603	MEDIUM	OTHER	7300	3	68165	728	Covers
ADESA	GREEN	No	6149	LARGE	FORD	6600	5	93346	1774	Alloy
ADESA	SILVER	Yes	6057	MEDIUM	CHRYSLER	6400	3	73963	1389	Covers
ADESA	SILVER	No	8113	SPECIALTY	CHRYSLER	10400	5	64839	1215	Alloy
ADESA	RED	No	6702	MEDIUM	GM	7100	4	63151	923	Covers
ADESA	MAROON	No	3320	MEDIUM	GM	4700	7	92782	1209	Alloy
ADESA	GREY	No	7708	SPECIALTY	CHRYSLER	9400	5	72592	1389	Alloy
ADESA	WHITE	No	2700	MEDIUM	GM	3900	8	88667	2712	Alloy
ADESA	RED	No	7860	MEDIUM	CHRYSLER	7500	2	50644	754	Covers
ADESA	SILVER	No	7785	LARGE	GM	8300	3	58384	1500	Alloy
ADESA	BLUE	No	8091	LARGE SUV	FORD	9500	6	80906	1113	Alloy
ADESA	WHITE	No	6793	SMALL SUV	OTHER	7935	5	59801	754	Alloy
ADESA	WHITE	No	6741	MEDIUM SU	FORD	9335	6	77178	1740	unkwnWheel
ADESA	GREY	No	3895	SMALL SUV	FORD	7100	8	79030	1220	unkwnWheel
ADESA	SILVER	Yes	6554	MEDIUM	OTHER	6700	4	61315	728	Alloy
ADESA	SILVER	No	2988	MEDIUM	GM	4700	9	92792	2651	Alloy
ADESA	GREY	No	5396	SPORTS	FORD	6600	6	82271	853	Alloy

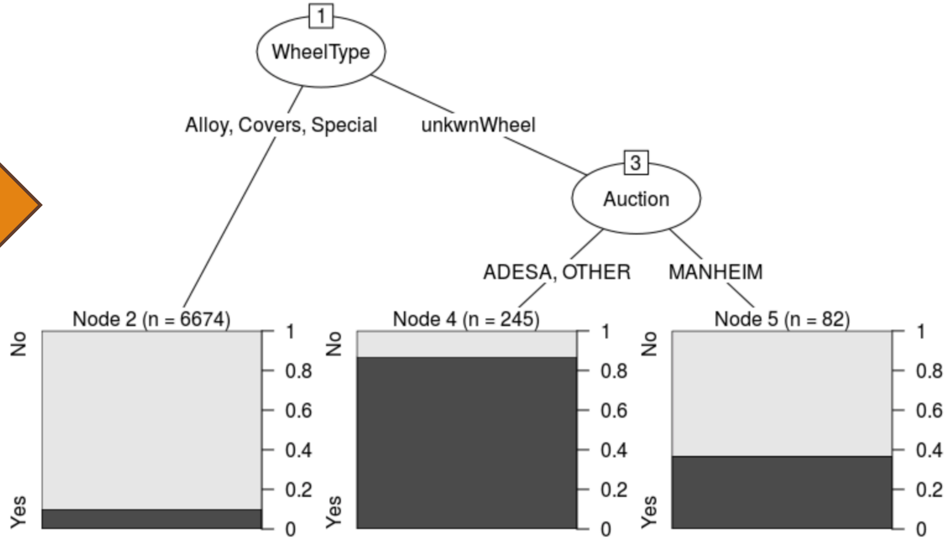
70% training data

30% testing data

Evaluate Decision Tree Model Performance

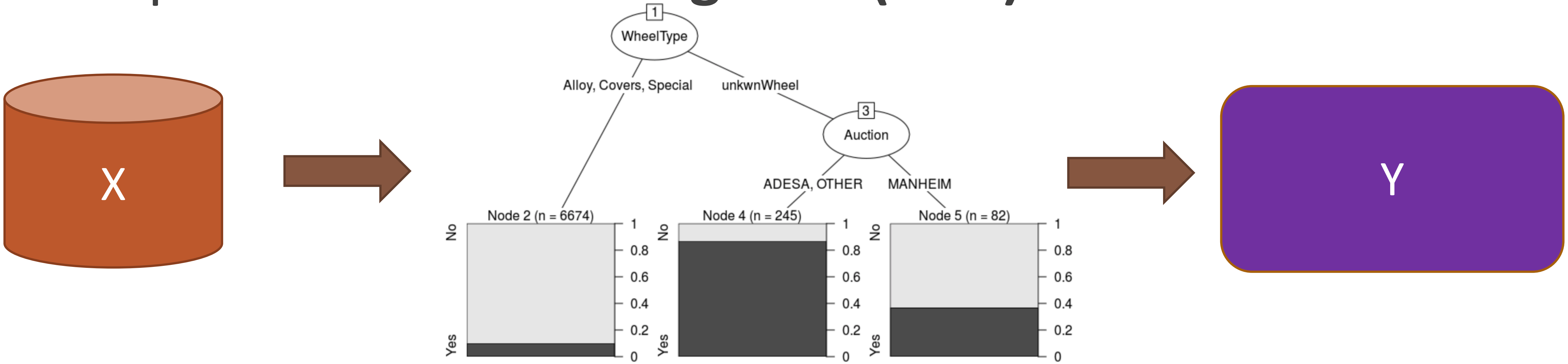
- Train decision tree on **training data (70%)**

Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	WHITE	No	2871	LARGE TRUC	FORD	5300	8	75419	869	Alloy
ADESA	GOLD	Yes	1840	VAN	FORD	3600	8	82944	2322	Alloy
ADESA	RED	No	8931	SMALL SUV	CHRYSLER	7500	4	57338	588	Alloy
ADESA	GOLD	No	8320	CROSSOVER	FORD	8500	5	55909	1169	Alloy
ADESA	GREY	No	11520	LARGE TRUC	FORD	10100	5	86702	853	Alloy
ADESA	SILVER	No	2659	COMPACT	GM	4100	7	73810	1455	Covers
ADESA	RED	No	4645	VAN	FORD	5600	5	85003	1633	Covers
ADESA	SILVER	No	4352	LARGE	GM	5900	5	88991	2152	Covers
ADESA	SILVER	No	5142	MEDIUM	GM	6600	5	80077	1373	Alloy
ADESA	MAROON	No	9983	MEDIUM	OTHER	7500	3	71952	1272	Alloy
ADESA	WHITE	No	4165	MEDIUM	OTHER	6200	4	23881	462	Covers
ADESA	GOLD	No	2422	VAN	GM	5100	9	83238	5392	Alloy
ADESA	SILVER	No	6603	MEDIUM	OTHER	7300	3	68165	728	Covers
ADESA	GREEN	No	6149	LARGE	FORD	6600	5	93346	1774	Alloy
ADESA	SILVER	Yes	6057	MEDIUM	CHRYSLER	6400	3	73963	1389	Covers
ADESA	SILVER	No	8113	SPECIALTY	CHRYSLER	10400	5	64839	1215	Alloy
ADESA	RED	No	6702	MEDIUM	GM	7100	4	63151	923	Covers
ADESA	MAROON	No	3320	MEDIUM	GM	4700	7	92782	1209	Alloy
ADESA	GREY	No	7708	SPECIALTY	CHRYSLER	9400	5	72592	1389	Alloy
ADESA	WHITE	No	2700	MEDIUM	GM	3900	8	88667	2712	Alloy
ADESA	RED	No	7860	MEDIUM	CHRYSLER	7500	2	50644	754	Covers
ADESA	SILVER	No	7785	LARGE	GM	8300	3	58384	1500	Alloy
ADESA	BLUE	No	8091	LARGE SUV	FORD	9500	6	80906	1113	Alloy
ADESA	WHITE	No	6793	SMALL SUV	OTHER	7935	5	59801	754	Alloy
ADESA	WHITE	No	6741	MEDIUM SU	FORD	9335	6	77178	1740	unkwnWheel
ADESA	GREY	No	3895	SMALL SUV	FORD	7100	8	79030	1220	unkwnWheel
ADESA	SILVER	Yes	6554	MEDIUM	OTHER	6700	4	61315	728	Alloy
ADESA	SILVER	No	2988	MEDIUM	GM	4700	9	92792	2651	Alloy
ADESA	GREY	No	5396	SPORTS	FORD	6600	6	82271	853	Alloy



Predictions with Decision Tree

- Make predictions on **testing data (30%)**



Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	RED		7860	MEDIUM	CHRYSLER	7500	2	50644	754	Covers
ADESA	SILVER		7785	LARGE	GM	8300	3	58384	1500	Alloy
ADESA	BLUE		8091	LARGE SUV	FORD	9500	6	80906	1113	Alloy
ADESA	WHITE		6793	SMALL SUV	OTHER	7935	5	59801	754	Alloy
ADESA	WHITE		6741	MEDIUM SU	FORD	9335	6	77178	1740	unkwnWheel
ADESA	GREY		3895	SMALL SUV	FORD	7100	8	79030	1220	unkwnWheel

IsBadBuy
No
No
No
No
Yes
Yes

Evaluate Decision Tree Model Performance

- Compare the **predictions** and **real values/actual value**

Predictions/predicted values

IsBadBuy
No
No
No
No
Yes
Yes

real values

IsBadBuy
No
No
No
No
No
No

Evaluate Decision Tree Model Performance

- Confusion Matrix
  - A **confusion matrix** is a table that categorizes predictions according to whether they match the actual value.
    - The most common performance measures consider the model's ability to discern one class versus all others. The class of interest is known as the **positive** class, while all others are known as **negative**.

Evaluate Decision Tree Model Performance

	Predicted Class Label		
True Class Label		**a**	**b**
	a	True Positive (TP)	False Negative (FN) (Type II error)
	b	False Positive (FP) (Type I error)	True Negative (TN)

Assume **a** is positive class and **b** is negative class

```
          pred
target    No   Yes
   No   2601    10
   Yes   302    86
```

- **True Positive (TP)**: Correctly classified as is positive class
- **True Negative (TN)**: Correctly classified as negative class
- **False Positive (FP)**: Incorrectly classified as positive class
- **False Negative (FN)**: Incorrectly classified as negative class

Evaluate Decision Tree Model Performance

	Predicted Class Label		
True Class Label		**a**	**b**
	a	True Positive (TP)	False Negative (FN) (Type II error)
	b	False Positive (FP) (Type I error)	True Negative (TN)

```
       pred
target   No  Yes
   No  2601   10
   Yes  302   86
```

- **a** is positive class
- **b** is negative class
- T (Total population) = TP+TN+FP+FN

- True class label is **a**= TP+FN
- Predicted class label is **a** = TP+FP
- True class label is **b** = FP+TN
- Predicted class label is **b** = FN+TN

Evaluate Decision Tree Model Performance

- **Accuracy** is the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- The **error rate** or the proportion of the incorrectly classified examples is specified as

$$\text{error rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = 1 - \text{accuracy}$$

Evaluate Decision Tree Model Performance

- Evaluate performances on positive/negative class
  - **Precision**
  - **recall**
  - **F-measure**

Evaluate Decision Tree Model Performance

- Evaluation Metrics: Precision
  - The **precision** is defined as the proportion of positive examples that are truly positive.
    - How many instances are predicted to the positive (**a**) class?
      - Predicted class label is **a** = TP+FP
    - How many of instances predicted as **a** actually belong to **a** class?
      - TP
  - Precision (**a** or positive) = TP/(TP+FP)=2601/(2601+302)
  - Precision (**b** or negative) = TN/(TN+FN) = 86/(86+10)

```
        pred
target   No  Yes
   No  2601   10
   Yes  302   86
```

Evaluate Decision Tree Model Performance

- Evaluation Metrics: Recall
  - **Recall** is the ability to correctly classify instances belonging to this class.
    - How many instances actually belong to the positive (**a**) class?
      - True class label is **a**= TP+FN
    - How many of instances predicted as **a** actually belong to **a** class?
      - TP
  - Recall (a or positive) = TP/(TP+FN) = 2601/(2601 + 10)
  - Recall (b or negative) = TN/(TN+FP) = 86/(86 + 302)

```
      pred
target   No  Yes
   No  2601   10
   Yes  302   86
```

Evaluate Decision Tree Model Performance

- F-measure
  - The harmonic mean of precision and recall.
  - It can be used as a single measure of overall performance on positive/negative class.
    - F-measure (**a**) = (2 x Precision(**a**) x Recall(**a**)) / (Precision(**a**) + Recall(**a**))
    - F-measure (**b**) = (2 x Precision(**b**) x Recall(**b**)) / (Precision(**b**) + Recall(**b**))

Evaluate Decision Tree Model Performance

Performance on the **training data**:

```
        pred
target    No  Yes
   No   6062   32
   Yes   694  213
```

Model's overall performance

Overall performance on Not bad buy class

Overall performance on bad bad buy class

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
89.63005	89.72765	86.93878	99.47489	23.48401	94.35019	36.97917

Performance on the **testing data**:

```
        pred
target    No  Yes
   No   2601   10
   Yes   302   86
```

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
89.59653	89.59697	89.58333	99.61700	22.16495	94.34168	35.53719