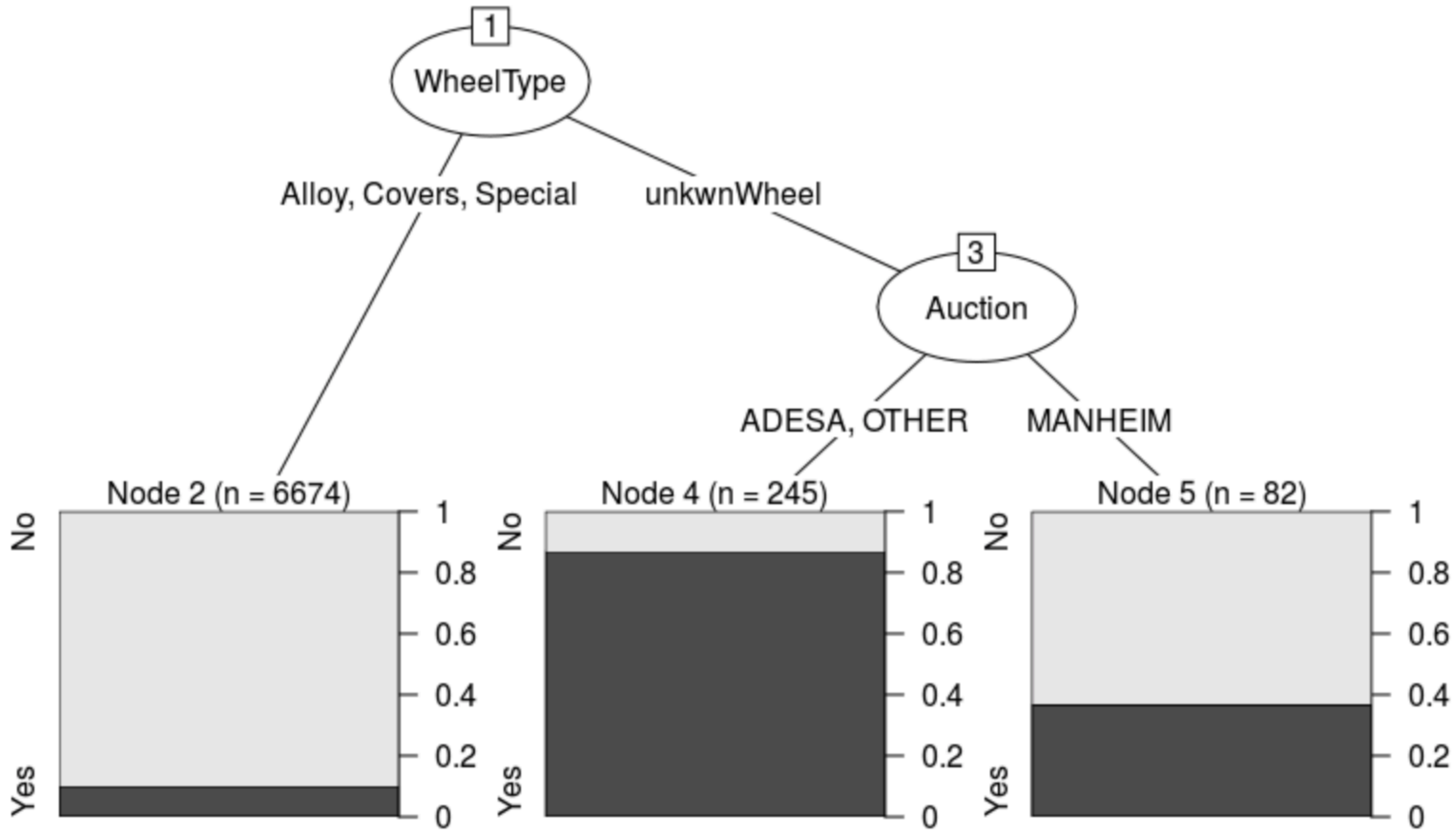
Lecture 4:Naïve Bayes

Decision Tree: Recap



<u>Classification rules</u>

- 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 : Recap

- 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

Evaluation: Recap

		Predicted Class Label	
		a	**b**
True Class Label	**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

Evaluation: Recap

	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

- **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

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

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

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

Evaluation: Recap

- Precision (**a**) = TP/(TP+FP)=2601/(2601+302)
- Precision (**b**) = TN/(TN+FN) = 86/(86+10)
- Recall (**a**) = TP/(TP+FN) = 2601/(2601 + 10)
- Recall (**b**) = TN/(TN+FP) = 86/(86 + 302)
- 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**))

target \ pred	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

Overview

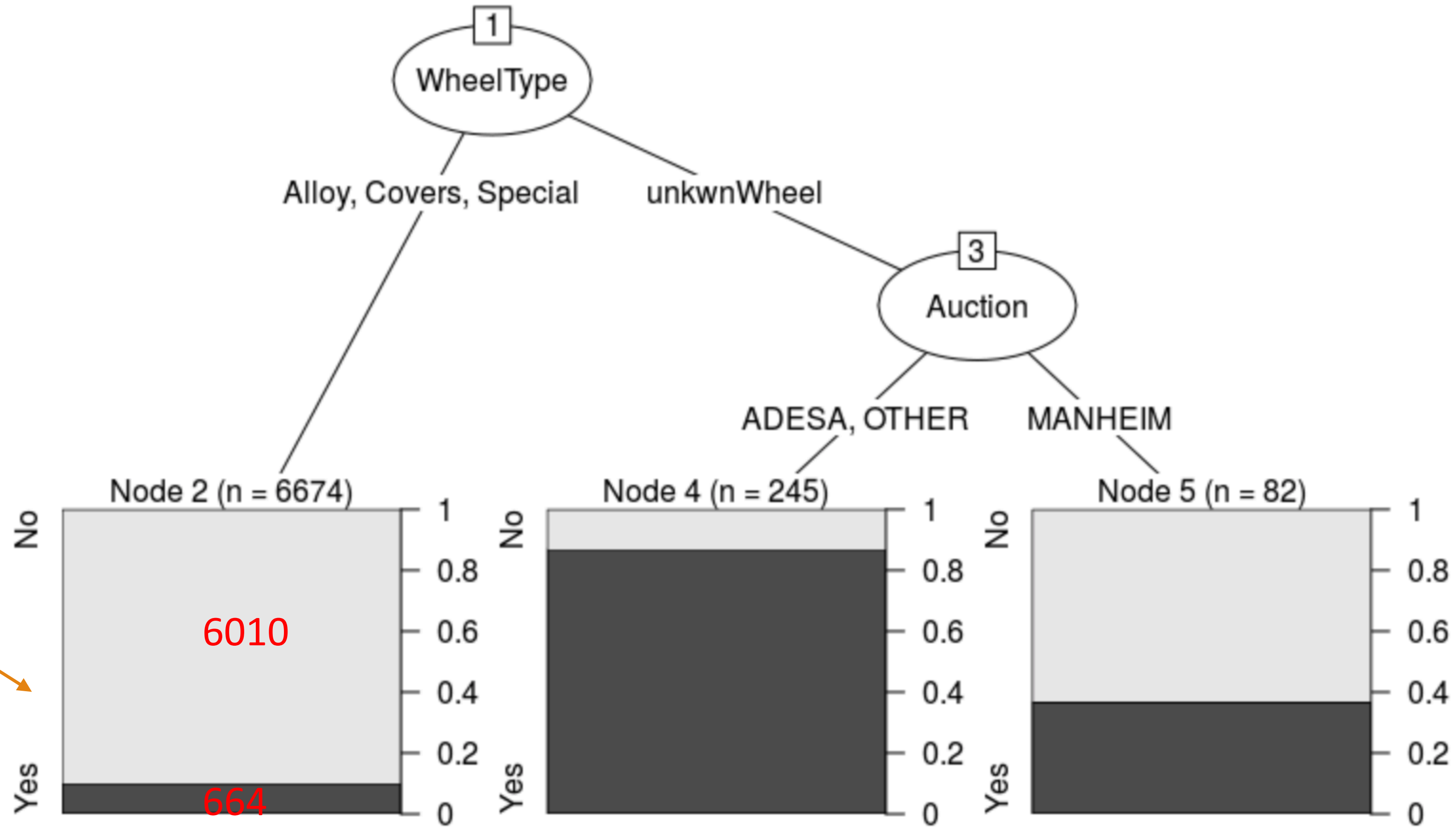
- Basic principles of probability
- Naïve Bayes Classification
- Overfitting and Its Avoidance
- Model Comparison

Background

- Model a classification rule directly
  - Decision Tree: rule-based model
- Make a probabilistic model of data within each class
  - Naïve Bayes: probabilistic model

Understanding Probability

P(IsBadBuy=No|WheelType = {Alloy, Covers, Special}) = 6010/(6010+664)



Understanding Probability

- Conditional and joint probability for random variables
  - Conditional probability $P(A|C)$: the probability of A is dependent (that is, conditional) on the value of C.
  - Joint probability: $P(A, C)$
  - Relationship: $P(A, C) = P(A|C)P(C)$
  - Independence: $P(A, C)=P(A)P(C)$, $P(A|C)=P(A)$, $P(C|A)=P(C)$

Understanding Probability

- Bayes' theorem, named after 18th-century British mathematician Thomas Bayes, is a mathematical formular for determining conditional probability.

$$P(C|A)=\frac{P(A, C)}{P(A)}=\frac{P(A|C)P(C)}{P(A)}$$

posterior

likelihood

prior

marginal probability

Understanding Probability

- You are planning a picnic today, but the morning is cloudy
  - Cloudy mornings are common (about 40% of days start cloudy)
  - And this is usually a dry month (only 3 of 30 days tend to be rainy, or 10%)
  - Rainy days start off cloudy
    - $\text{P(Rain|Cloud)} = \frac{\text{P(Cloud|Rain) P(Rain)}}{\text{P(Cloud)}} = \frac{1*0.1}{0.4} = 0.25$

Naïve Bayes

- The **Naive Bayes** algorithm describes a simple method to apply Bayes' theorem to classification problems.
  - The Naive Bayes algorithm is named as such because it makes some "naive" assumptions about the data.
    - Naive Bayes assumes that all of the features in the dataset are equally important and independent. (strong assumption)
    - However, in most cases when these assumptions are violated, Naive Bayes still performs fairly well.
- Naive assumptions + Bayes’ theorem

Understanding Probability

- Prediction with one predictor
  - aggregate(WheelType~IsBadBuy, summary,data = carAuction)

	IsBadBuy	WheelType.Alloy	WheelType.Covers	WheelType.Special	WheelType.unkwnWheel	Total
1	No	4340	4171	86	108	8705
2	Yes	581	365	11	338	1295

$$\text{P(C|A)} = \frac{\text{P(A|C)P(C)}}{\text{P(A)}}$$

posterior

likelihood

marginal probability

prior

  - $\text{P(IsBadBuy=Yes| WheelType=unkwnWheel)} = \dfrac{\dfrac{338}{1295} * \dfrac{1295}{1295+8705}}{\dfrac{108+338}{1295+8705}}$

Naïve Bayes

- More variables
  - Consider each attribute and class label as random variables
  - Given a record/instance with attributes (A_1, A_2, A_3, … A_n)
    - Goal is to predict the value of C
    - Specifically, we want to find the value of C that maximizes $P(C| A_1, A_2, A_3, \ldots A_n)$
  - Can we estimate $P(C| A_1, A_2, A_3, \ldots A_n)$ directly from data?
    - P(IsBadBuy=Yes| WheelType=unkwnWheel, Auction=OTHER, Color=Red)

Naïve Bayes

- Compute the posterior probability $P(C| A_1, A_2, A_3, \ldots A_n)$ for all values of C using the Bayes' theorem

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

Naïve Bayes

- How to estimate $P(A_1, A_2, \ldots A_n | C)$
  - Select all instances of class C in training set
  - Count all possible combinations $A_1, A_2, \ldots A_n$
- However,
  - Not all combinations are present
- Hence:
  - Additional assumptions on the distribution
  - Conditional independence

Naïve Bayes

- Assume ***conditional independence*** among attributes A_i when class is given

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

$$= \frac{\left(\prod_{i=1}^{n} P(A_i \mid C) \right) P(C)}{P(A_1 A_2 \ldots A_n)}$$

conditional independence assumption

$P(A_1, A_2, \ldots A_n | C) = P(A_1|C)P(A_2|C)\ldots P(A_n|C)$

Naïve Bayes

- $$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$ Bayes' theorem

- $$= \frac{\left(\prod_{i=1}^{n} P(A_i \mid C) \right) P(C)}{P(A_1 A_2 \ldots A_n)}$$ conditional independence assumption

- $$\propto \left(\prod_{i=1}^{n} P(A_i \mid C) \right) P(C)$$ The final prediction depends on $P(A_i|C)$ and P(C)

$$\frac{P(A_i, C)}{P(C)}$$

Naïve Bayes

WheelType	Auction	IsBadBuy
Alloy	OTHER	Yes
Special	ADESA	No
Alloy	MANHEIM	No
unkwnWheel	OTHER	No
unkwnWheel	OTHER	Yes

Prediction for (WheelType=unkwnWheel, Auction=OTHER)

$$\left(\prod_{i=1}^{n} P(A_i \mid C)\right) P(C)$$

- P(IsBadBuy = Yes|WheelType=unkwnWheel, Auction=OTHER) ∝ P(WheelType=unkwnWheel| IsBadBuy = Yes) *P(Auction=OTHER| IsBadBuy = Yes)*P(C) = 0.5 * 1 * 0.4 = 0.2
- P(IsBadBuy = No|WheelType=unkwnWheel, Auction=OTHER) ∝ P(WheelType=unkwnWheel| IsBadBuy = No) *P(Auction=OTHER| IsBadBuy = No)*P(C) = 0.333 * 0.333 * 0.6 = 0.0665

- IsBadBuy = Yes (40%; 2 instances)

WheelType:	Alloy	1
	Special	0
	unkwnWheel	1
Auction:	ADESA	0
	MANHEIM	0
	OTHER	2

- IsBadBuy = No (60%; 3 instances)

WheelType:	Alloy	1
	Special	1
	unkwnWheel	1
Auction:	ADESA	1
	MANHEIM	1
	OTHER	1

Naïve Bayes

- Prediction for (WheelType=unkwnWheel, Auction=OTHER)

$$\left(\prod_{i=1}^{n} P(A_i \mid C)\right) P(C)$$

- IsBadBuy = Yes: P(unkwnWheel|Yes)*P(OTHER |Yes)*P(Yes)=0.5 * 1 * 0.4 = 0.2
- IsBadBuy = No: P(unkwnWheel|No)*P(OTHER |No)*P(No)= 0.333 * 0.333 * 0.6 = 0.0665

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

$$= \frac{\left(\prod_{i=1}^{n} P(A_i \mid C)\right) P(C)}{P(A_1 A_2 \ldots A_n)}$$

P(Yes|WheelType=unkwnWheel, Auction=OTHER)= $\frac{0.2}{0.2665}$

P(No|WheelType=unkwnWheel, Auction=OTHER)= $\frac{0.0665}{0.2665}$

marginal probability

$P(A_1, A_2, \ldots A_n) = \sum_C P(A_1, A_2, \ldots An, C) = \sum_C P(A_1, A_2, \ldots An \mid C) P(C) = \sum_c \left(\prod_{i=1}^{n} P(A_i \mid C)\right) P(C) = 0.2 + 0.0665 = 0.2665$

Naïve Bayes

WheelType	Auction	IsBadBuy
Alloy	OTHER	Yes
Special	ADESA	No
Alloy	MANHEIM	No
unkwnWheel	OTHER	No
unkwnWheel	OTHER	Yes

Prediction for (WheelType=Special, Auction=OTHER)

$$\left(\prod_{i=1}^{n} P(A_i \mid C)\right) P(C)$$

zero value causes the posterior to be zero

- P(IsBadBuy = Yes| WheelType=Special, Auction=OTHER) ∝ P(WheelType= Special | IsBadBuy = Yes) *P(Auction=OTHER| IsBadBuy = Yes)*P(C) = 0 * 1 * 0.4 = 0
- P(IsBadBuy = No|WheelType= Special, Auction=OTHER) ∝ P(WheelType= Special | IsBadBuy =No) ∗P(Auction=OTHER| IsBadBuy =No)∗P(C) =0.333 * 0.333 * 0.6 = 0.0665

- IsBadBuy = Yes (40%; 2 instances)

WheelType:	Alloy	1
	Special	0
	unkwnWheel	1
Auction:	ADESA	0
	MANHEIM	0
	OTHER	2

- IsBadBuy = No (60%; 3 instances)

WheelType:	Alloy	1
	Special	1
	unkwnWheel	1
Auction:	ADESA	1
	MANHEIM	1
	OTHER	1

Naïve Bayes

- **Laplace estimator/Laplace smoothing**
  - The Laplace estimator essentially adds a small number to each of the counts, which ensures that each feature has a nonzero probability of occurring with each class.
    - Typically, the Laplace estimator is set to 1, which ensures that each class-feature combination is found in the data at least once.
    - In practice, given a large enough training dataset, this Laplace estimator is unnecessary and the value of 1 is almost always used.
    - Laplace smoothing is useful especially when the dataset is small

Naïve Bayes

WheelType	Auction	IsBadBuy
Alloy	OTHER	Yes
Special	ADESA	No
Alloy	MANHEIM	No
unkwnWheel	OTHER	No
unkwnWheel	OTHER	Yes

Prediction for (WheelType=Special, Auction=OTHER)

$$\left(\prod_{i=1}^{n} P(A_i \mid C)\right) P(C)$$

- P(IsBadBuy = Yes| WheelType=Special, Auction=OTHER) ∝ P(WheelType= Special | IsBadBuy = Yes) *P(Auction=OTHER| IsBadBuy = Yes)*P(C) = 0.2 * 0.6 * 0.4 = 0.048
- P(IsBadBuy = No|WheelType= Special, Auction=OTHER) ∝ P(WheelType= Special | IsBadBuy =No) ∗P(Auction=OTHER| IsBadBuy =No)∗P(C) = ∝ 0.333 * 0.333 * 0.6 = 0.0665

- IsBadBuy = Yes (40%; 2 instances)

WheelType:	Alloy	2
	Special	1
	unkwnWheel	2
Auction:	ADESA	1
	MANHEIM	1
	OTHER	3

- IsBadBuy = No (60%; 3 instances)

WheelType:	Alloy	2
	Special	2
	unkwnWheel	2
Auction:	ADESA	2
	MANHEIM	2
	OTHER	2

Naïve Bayes

- $$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$ Bayes' theorem
- $$= \frac{\left(\prod_{i=1}^{n} P(A_i \mid C) \right) P(C)}{P(A_1 A_2 \ldots A_n)}$$ conditional independence assumption
- $$\propto \left(\prod_{i=1}^{n} P(A_i \mid C) \right) P(C)$$ The final prediction depends on $\mathrm{P}(A_i|\mathrm{C})$ and P(C)

Naïve Bayes

- P(C)
  - C is the target variable (categorical variable)
  - P(C) is easy to calculate
    - P(IsBadBuy = Yes) = 0.4, P(IsBadBuy = No) = 0.6
- $\mathrm{P}(A_i|\mathrm{C})$
  - If A_i is a categorical variable
    - $\text{P(Auction=OTHER|IsBadBuy=Yes)} = \dfrac{\text{P(Auction=OTHER,IsBadBuy=Yes)}}{\text{P(C=Yes)}}$
  - A_i is a numeric variable?—Probability density estimation

Proportion of instances that have Auction = OTHER, and IsBadBuy= Yes

Proportion of instances that have IsBadBuy= Yes

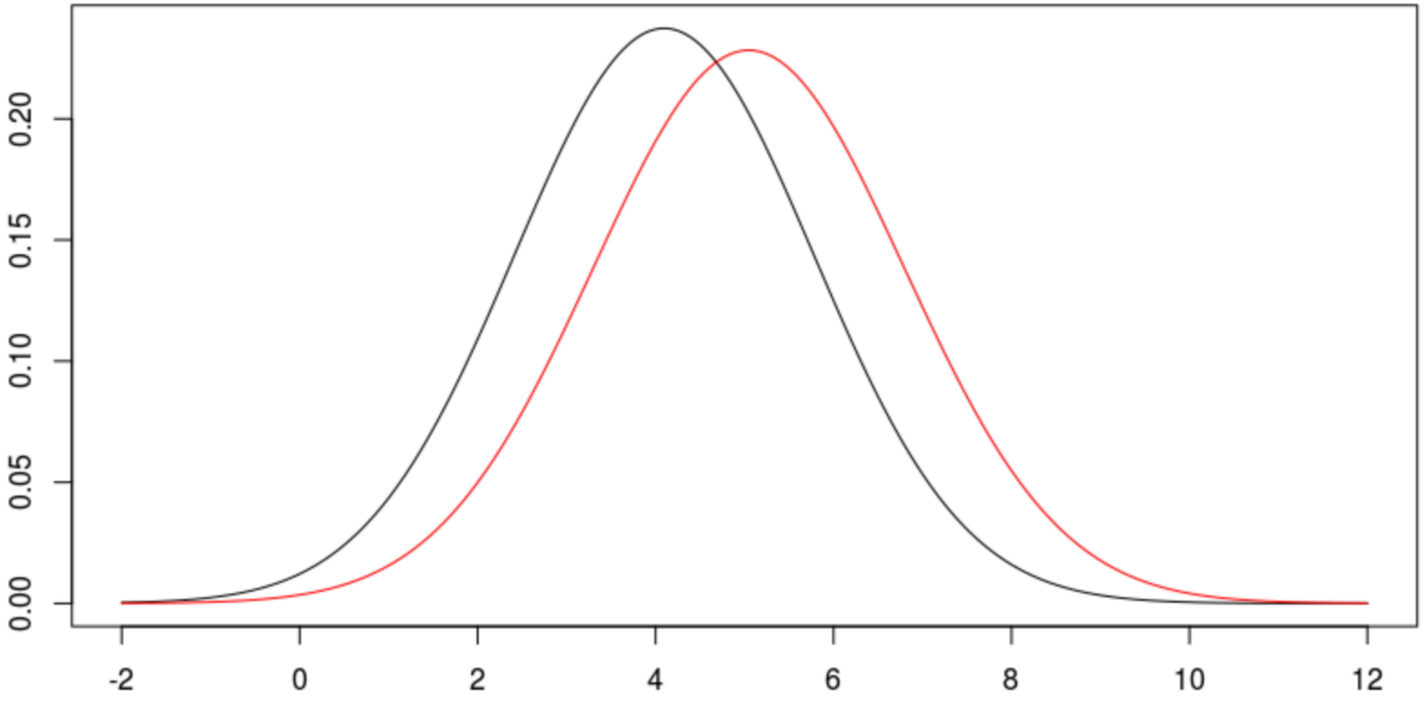
Naïve Bayes

- $P(A_i|C)$: Numeric variables with Naive Bayes
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once the probability distribution is known, can use it to estimate the conditional probability $P(A_i|C)$

Naïve Bayes

- For variable VehicleAge
  - If IsBadBuy = No,
    - Mean 4.095996, standard deviation 1.680877
  - If IsBadBuy = Yes,
    - Mean 5.047409, standard deviation 1.747582

```
          VehicleAge
Y              [,1]     [,2]
  No   4.095996 1.680877
  Yes  5.047409 1.747582
```



Naïve Bayes

WheelType	Auction	VehicleAge	IsBadBuy
Alloy	OTHER	3	Yes
Special	ADESA	5	No
Alloy	MANHEIM	4	No
unkwnWheel	OTHER	3	No
unkwnWheel	OTHER	6	Yes

Prediction for (WheelType=unkwnWheel, Auction=OTHER, VehicleAge=5)

- P(IsBadBuy = Yes|WheelType= unkwnWheel, Auction=OTHER, VehicleAge=5) ∝ P(WheelType= unkwnWheel|IsBadBuy = Yes) *P(Auction=OTHER|IsBadBuy = Yes)*P(VehicleAge=5|IsBadBuy = Yes)*P(C) = 0.5 * 1 * 0.228 *0.4 = 0.0456
- P(IsBadBuy = No|WheelType= unkwnWheel, Auction=OTHER, VehicleAge=5) ∝ P(WheelType= unkwnWheel|IsBadBuy = No) *P(Auction=OTHER|IsBadBuy = No)*P(VehicleAge=5|IsBadBuy = No)*P(C) = 0.333 * 0.333 * 0.205 * 0.6 = 0.0136

- IsBadBuy = Yes (40%; 2 instances)

WheelType:	Alloy	1
	Special	0
	unkwnWheel	1
Auction:	ADESA	0
	MANHEIM	0
	OTHER	2

- IsBadBuy = No (60%; 3 instances)

WheelType:	Alloy	1
	Special	1
	unkwnWheel	1
Auction:	ADESA	1
	MANHEIM	1
	OTHER	1

```
      VehicleAge
Y         [,1]     [,2]
  No  4.095996 1.680877
  Yes 5.047409 1.747582
```

Naïve Bayes

- Learning the model
  - For each class of C:
    - Estimate the prior P(C)
    - For each attribute A, for each attribute value v of A:
      - Estimate P(A=v |C)
- Applying the model
  - Given an example (v_1, v_2, v_3, … v_n)
  - Pick the class C that maximizes

$$\left(\prod_{i=1}^{n} P(A_i = v_i \mid C)\right) P(C)$$

Naïve Bayes Evaluation

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

Naïve Bayes Evaluation

- Train Naïve Bayes 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

Calculate frequency/ count for each categorical variable

Generate distribution for each numeric variable

- IsBadBuy = Yes (40%; 2 instances)

WheelType:	Alloy	1
	Special	0
	unkwnWheel	1
Auction:	ADESA	0
	MANHEIM	0
	OTHER	2

Prior

- IsBadBuy = No (60%; 3 instances)

WheelType:	Alloy	1
	Special	1
	unkwnWheel	1
Auction:	ADESA	1
	MANHEIM	1
	OTHER	1

```
     VehicleAge
Y         [,1]     [,2]
  No  4.095996 1.680877
  Yes 5.047409 1.747582
```

Naïve Bayes Evaluation

- Make predictions on **testing data (30%)** and **training data (70%)**
- $$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$ Bayes' theorem
- $$= \frac{\left(\prod_{i=1}^{n} P(A_i \mid C)\right) P(C)}{P(A_1 A_2 \ldots A_n)}$$ conditional independence assumption
- $$\propto \left(\prod_{i=1}^{n} P(A_i \mid C)\right) P(C)$$ The final prediction depends on $\mathrm{P}(A_i|\mathrm{C})$ and P(C)
  - A_i is a categorical variable: Find counts for $\mathrm{P}(A_i, \mathrm{C})$ and P(C)
  - A_i is a numeric variable: Probability density estimation

Naïve Bayes Evaluation

- 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

Naïve Bayes Evaluation

Performance on the **training data**:

Model's overall performance

Overall performance on NOT bad buy class

Overall performance on bad buy class

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
87.01614	90.53948	49.83498	95.01149	33.29658	92.72160	39.92069

Performance on the **testing data**:

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
86.26209	89.62162	44.64286	95.25086	25.77320	92.35054	32.67974

Generalization and Overfitting

One of the most important fundamental notions of data mining is that of overfitting and generalization.

- Generalization is the property of a model or modeling process, whereby the model applies to data that were not used to build the model.
- Overfitting is the tendency of data mining procedures to tailor models to the training data, at the expense of generalization to previously unseen data points.

Generalization and Overfitting

- Model performance
  - In-sample (training): evaluated using training data
  - Out-of-sample (i.e., generalization or test): evaluated using hold-out data
- Model generalization
  - Generalizable model – In-sample and out-of-sample performance levels are sufficiently similar
  - Non-generalizable model - model overfitting training data
  - Non-generalizable models don’t give accurate, reliable model performance estimations.

Generalization and Overfitting

Causes for Overfitting:

- Training data is not a good representation of testing (new) data
  - Insufficient training data
  - Noises in data: inconsistent class labels for the same values in feature set (input attributes)
  - Outliers in data: the number of samples with a given combination of class labels and feature values is small.
- An algorithm's inability to avoid overfitting noises/outliers or to train generalizable models via small amounts of training data
  - Complex model

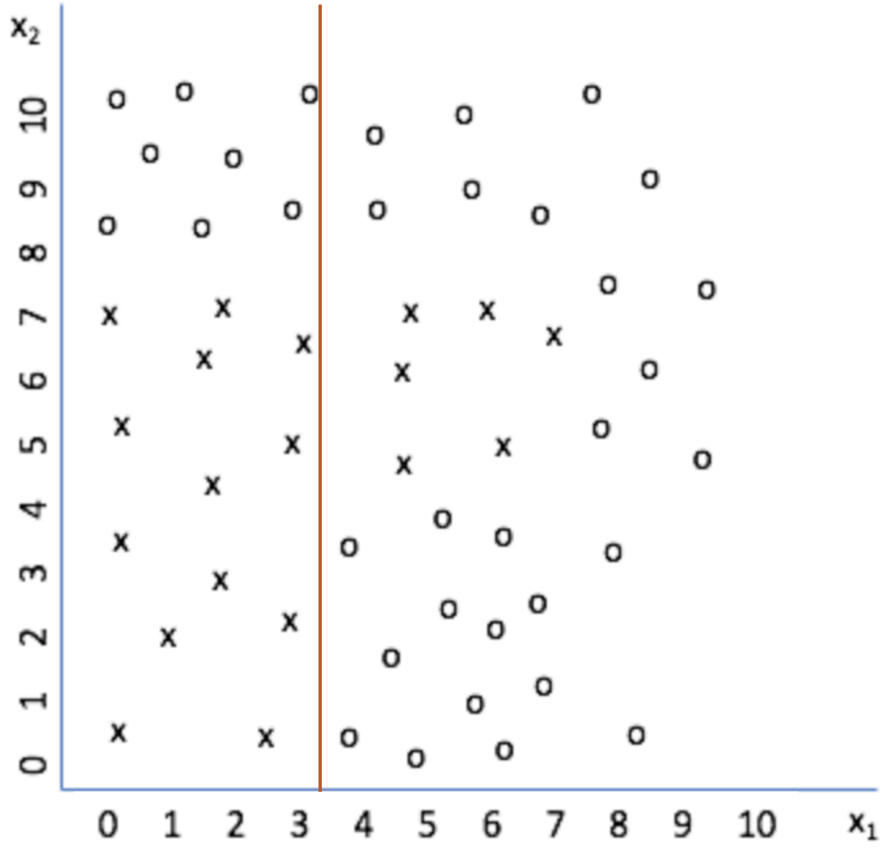
Generalization and Overfitting

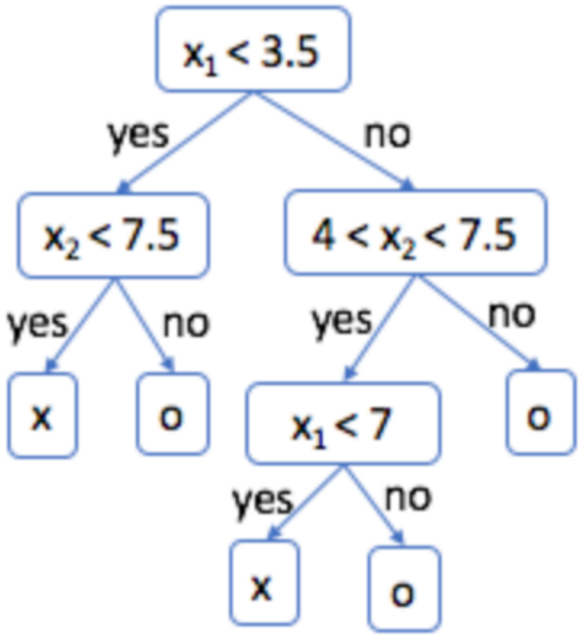
Avoidance of Overfitting

- Data strategies
  - Secure sufficient data
  - Identify and handle potential outliers and noises
- Evaluation strategies
  - Identify overfitting – Hold-out evaluations
- Model strategies
  - Select proper algorithm and manage model complexity
    - Compare different algorithms
    - Lower model complexity via method-specific parameters

Generalization and Overfitting

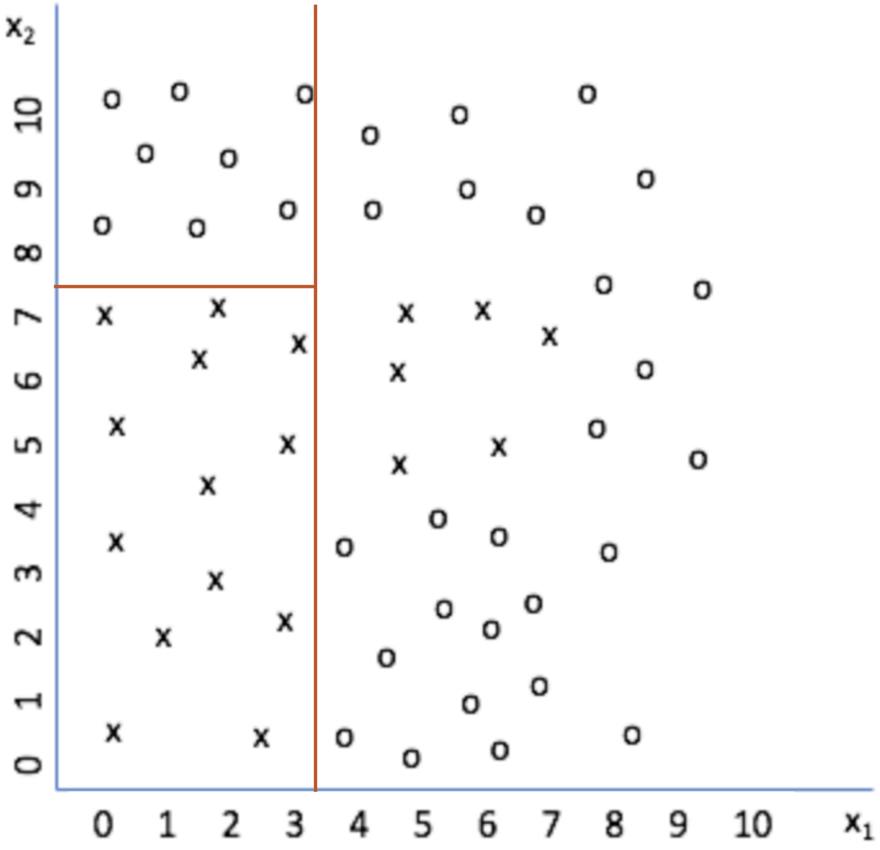
- Decision Tree

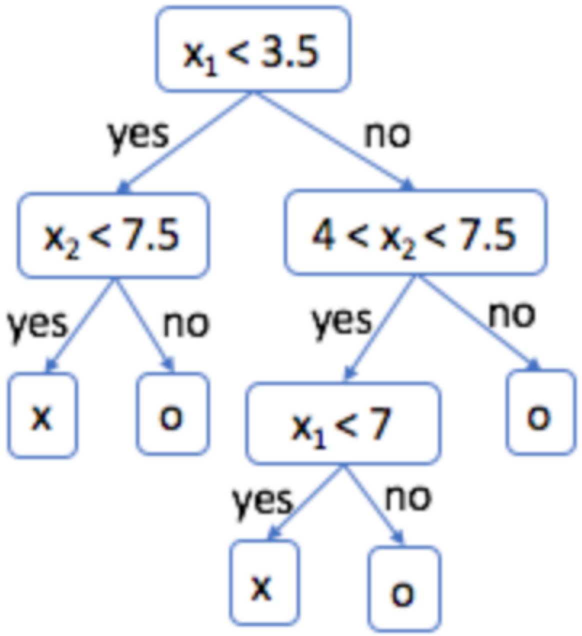




Generalization and Overfitting

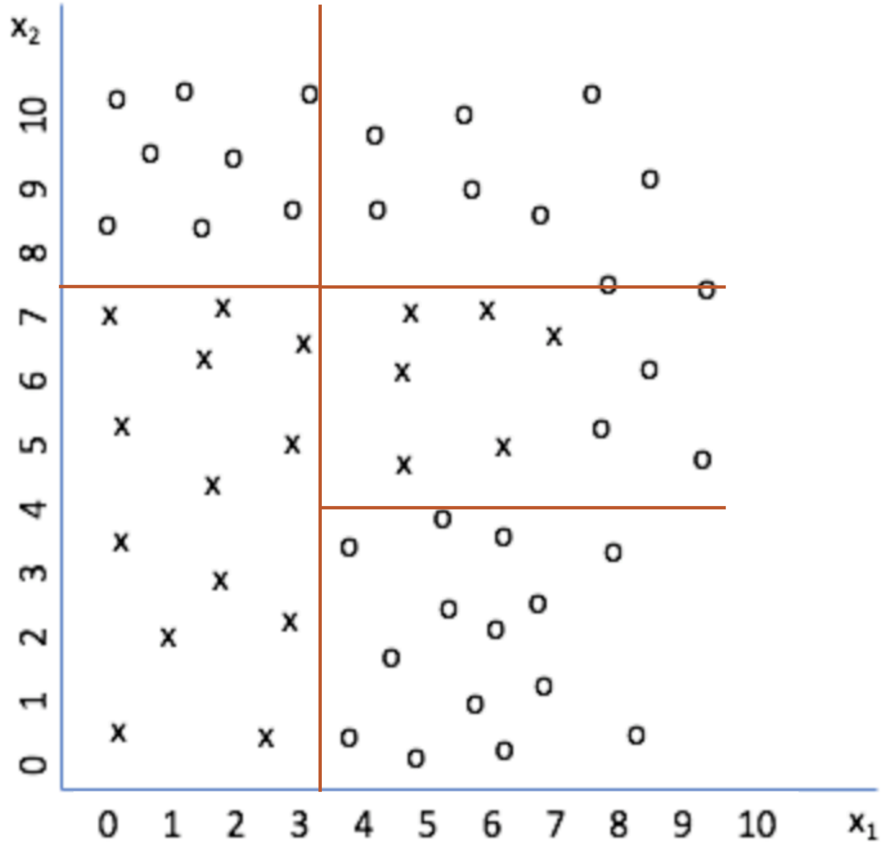
- Decision Tree

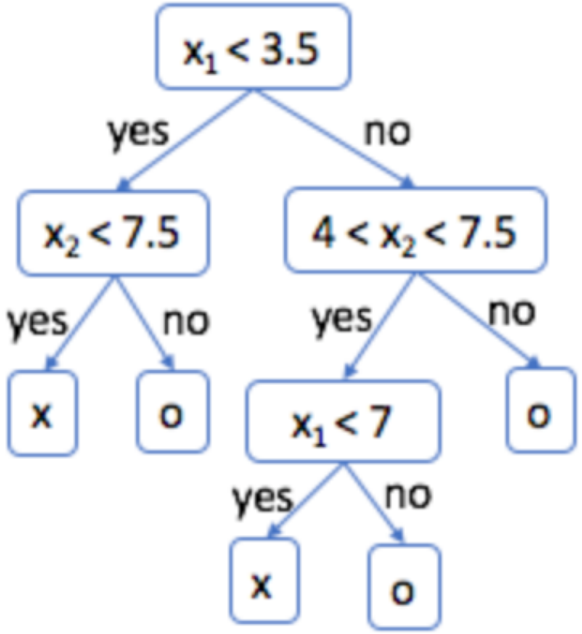




Generalization and Overfitting

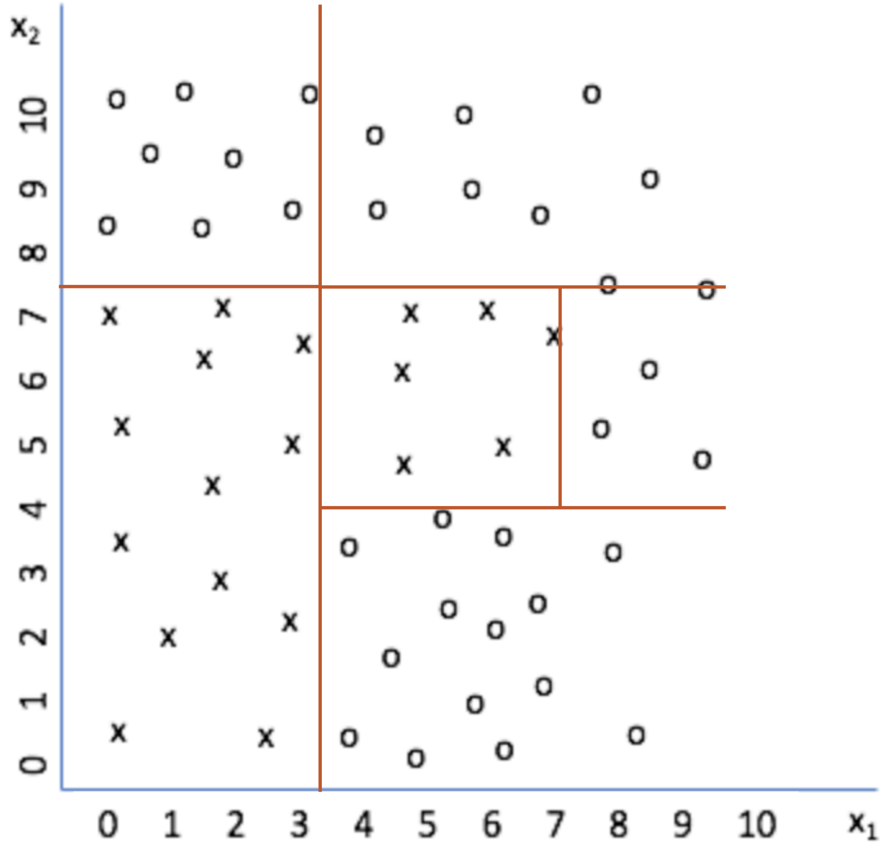
- Decision Tree

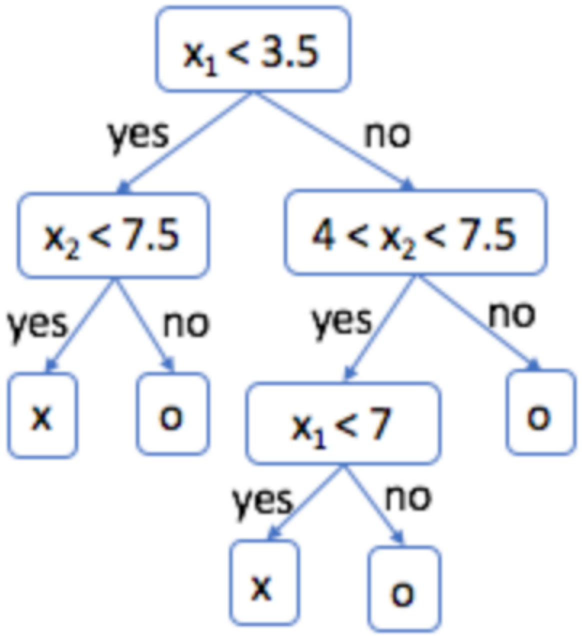




Generalization and Overfitting

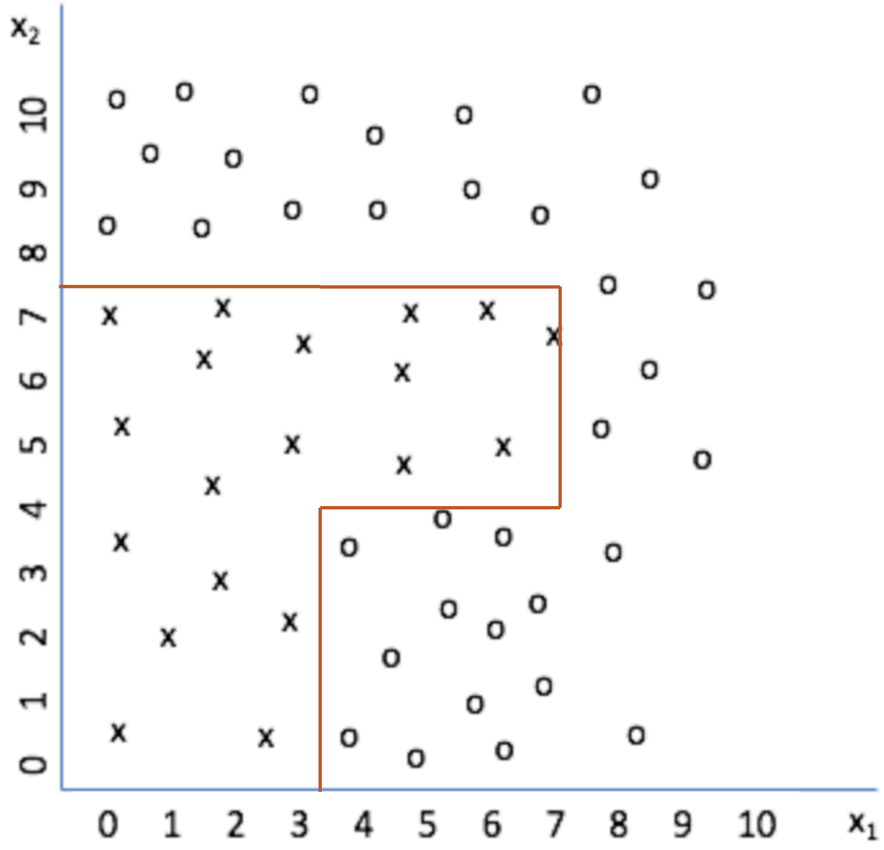
- Decision Tree

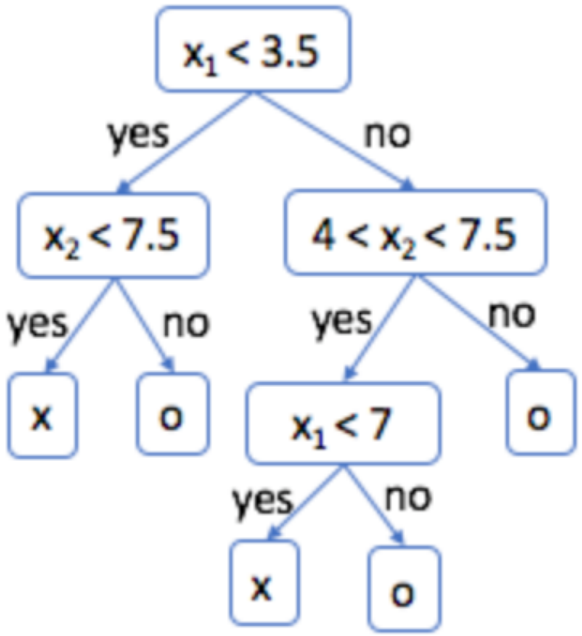




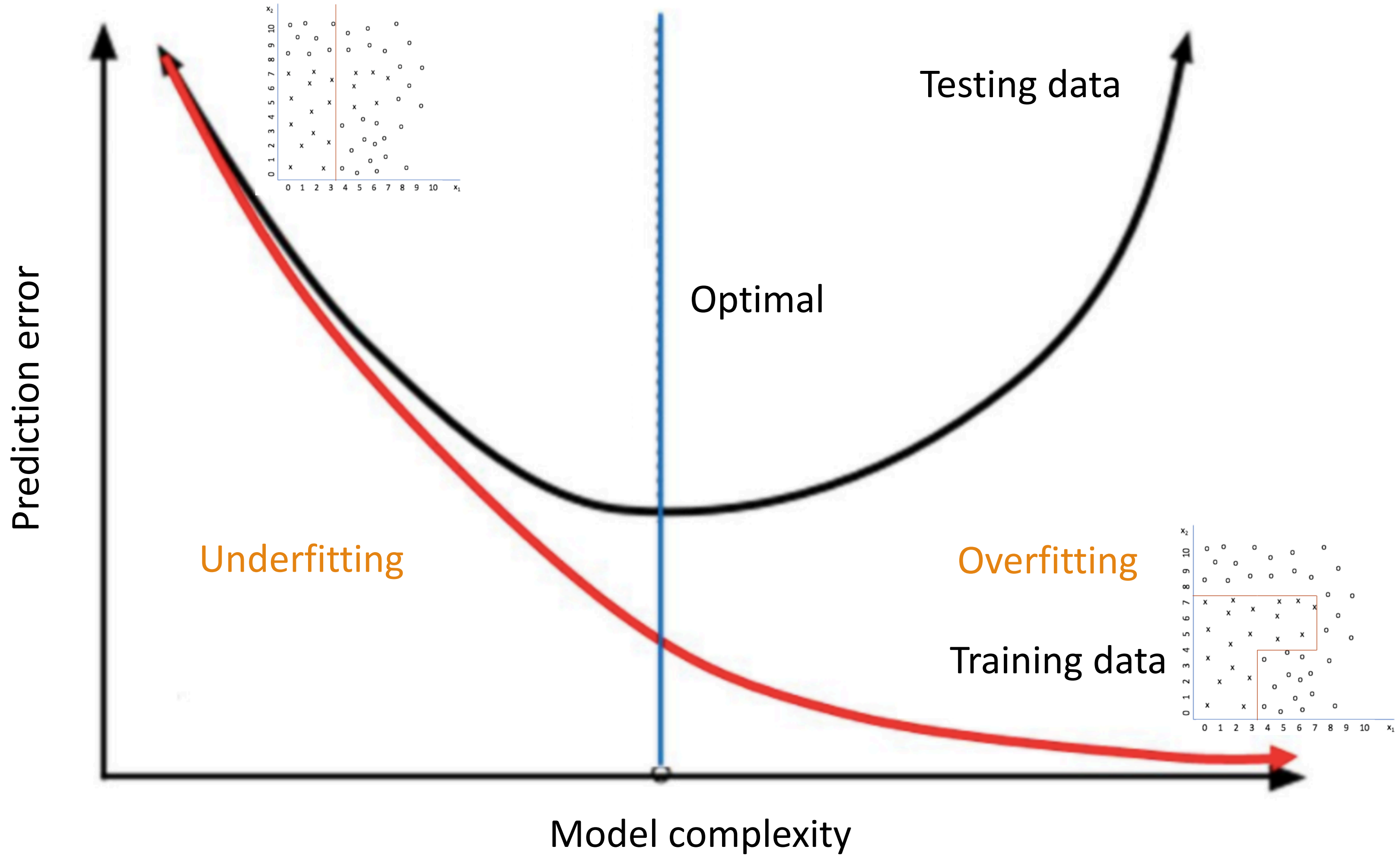
Generalization and Overfitting

- Decision Tree





Generalization and Overfitting



Generalization and Overfitting

- Complex tree
  - Tree size
    - Many decision and leaf nodes
  - Tree levels (depth)
    - How many nodes will be visited before reaching a leaf? (i.e., the length of a path)
    - A long tree path -> a complex rule of a sequence of many conjunctive conditions
- Complex trees – Some leaves may have very few instances or potentially outliers/noises (rare instances)

Principle of Occam's Razor

"Among competing hypotheses, the one with fewest assumptions should be selected", William of Occam, 13th Century

Complexity	Train Error	Test Error
Simple	0.23	0.24
Moderate	0.12	0.15
Complex	0.07	0.15
Super complex	0	0.18

When two trees have similar classification error on the validation (test) set, pick the simpler one.

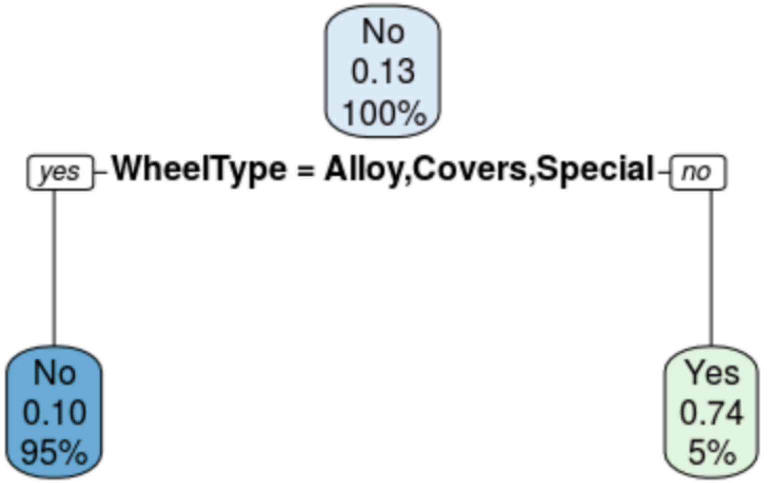
Generalization and Overfitting

Avoidance of Overfitting

- **Identifying overfitting**: comparing the model **performance** on **training** and **testing** data

Generalization and Overfitting

- Comparing the model **performances** on **training** and **testing** data
  - Max depth = 1

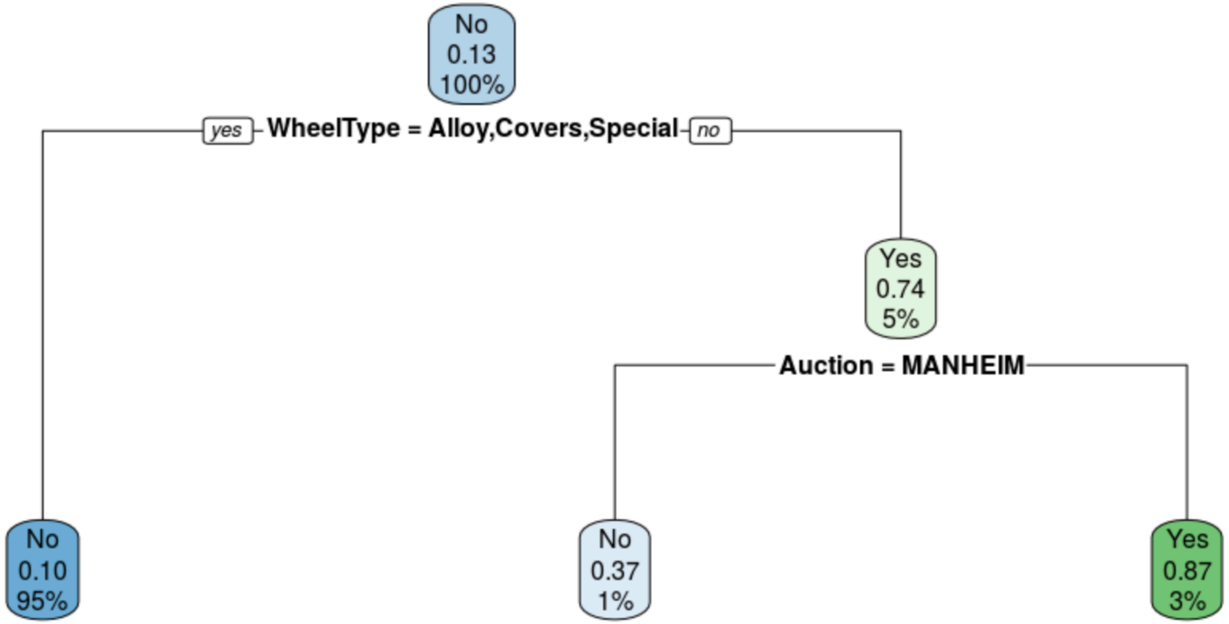


ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
89.31581	90.05094	74.31193	98.62160	26.79162	94.14160	39.38412

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
89.42981	89.82639	79.83193	99.08081	24.48454	94.22692	37.47535

Generalization and Overfitting

- Comparing the model **performances** on **training** and **testing** data
  - Max depth = 2

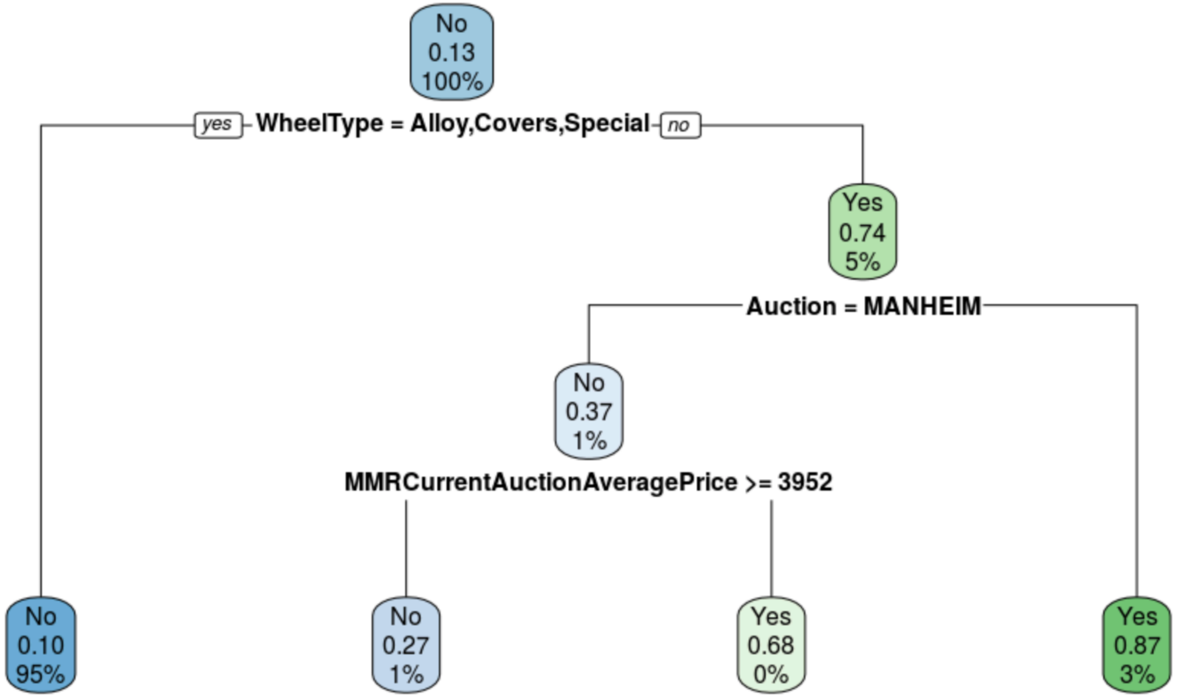


ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
89.63005	89.72765	86.93878	99.47489	23.48401	94.35019	36.97917

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

Generalization and Overfitting

- Comparing the model **performance** on **training** and **testing** data
  - Max depth = 3

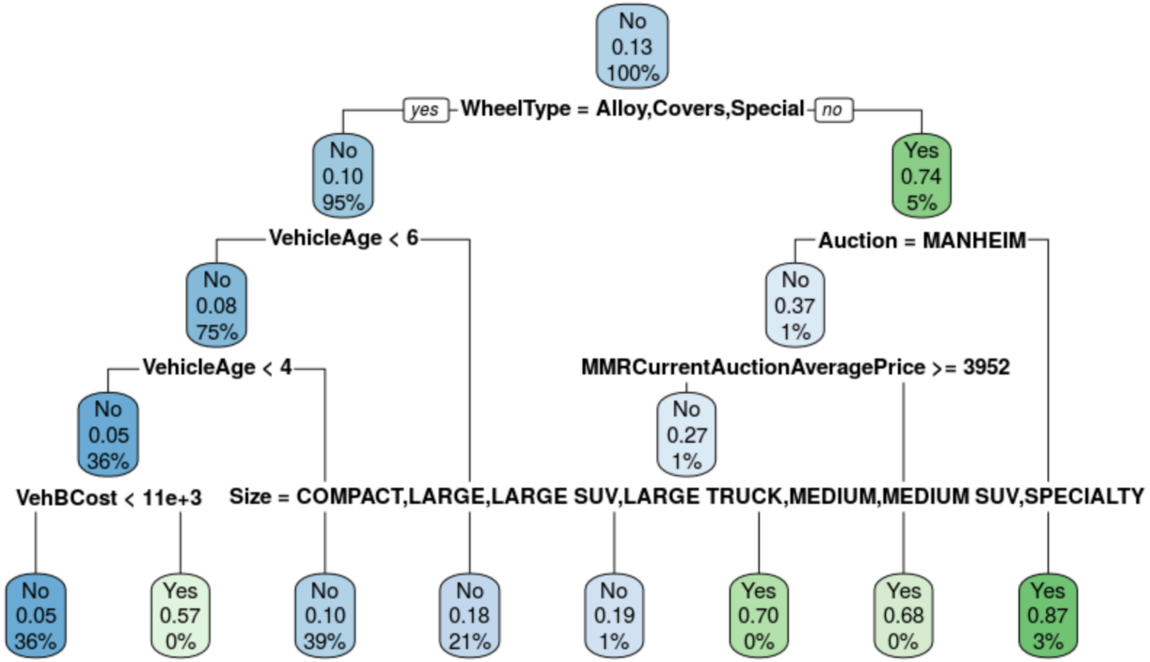


ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
89.73004	89.89164	85.60606	99.37644	24.91731	94.39638	38.59949

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
89.59653	89.67898	87.25490	99.50211	22.93814	94.33551	36.32653

Generalization and Overfitting

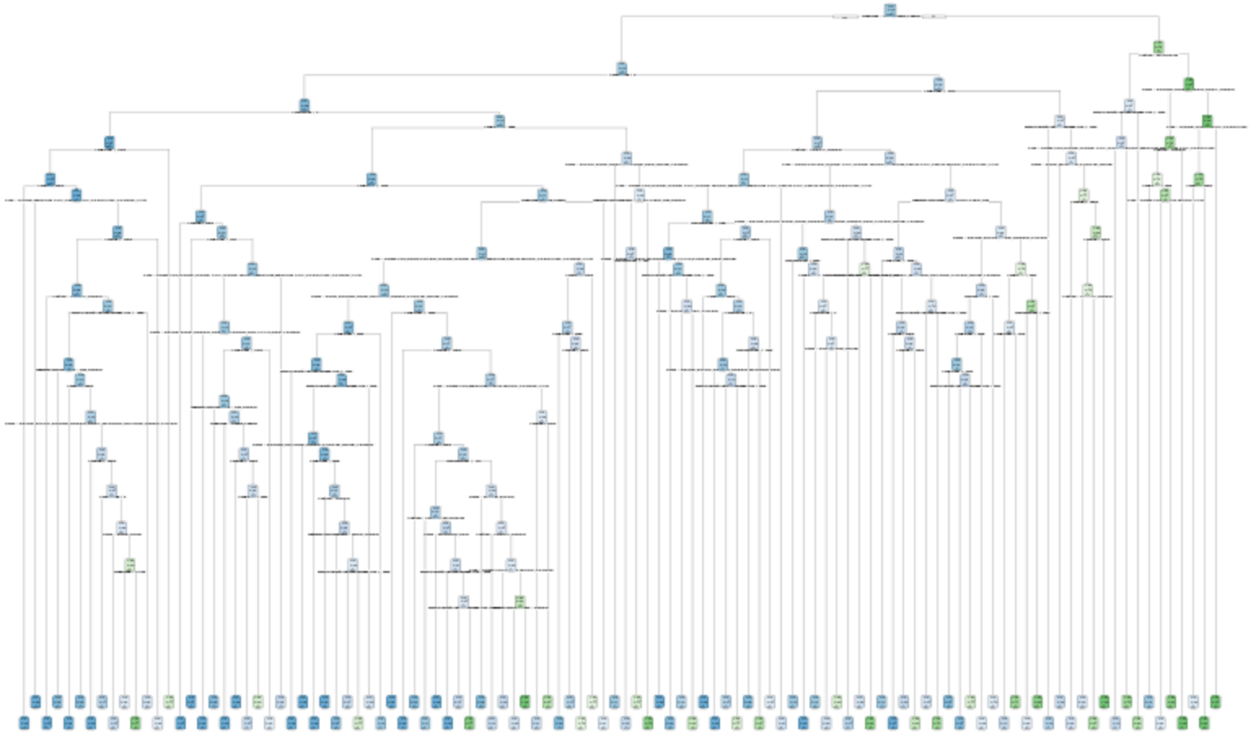
- Comparing the model **performance** on **training** and **testing** data
  - Max depth = 4




```
     ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
89.80146   90.02976   84.34164   99.27798   26.13010   94.42797   39.89899
     ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
89.56319   89.75779   84.40367   99.34891   23.71134   94.31013   37.02213
```

Generalization and Overfitting

- Comparing the model **performance** on **training** and **testing** data
  - Max depth = 15



ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
91.14412	91.97853	79.83368	98.40827	42.33738	95.08483	55.33141

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
86.92898	89.75278	49.03846	95.94025	26.28866	92.74343	34.22819

Generalization and Overfitting

- Optimal tree complexity
  - Decision tree with max depth = 2
  - Highest accuracy on testing data– lowest test/validation error
- Best performing model?
  - Best overall performance: Decision tree with max depth = 2
  - Best performance on minority class: Decision tree with max depth = 1
  - The best model is depend on business scenario

Model Comparison

- Relative to benchmarks and baselines
  - Random (coin-tossed) – e.g. 50% for 2 classes
  - Majority-rule: all instances are classified to the majority class
  - Other methods
    - Other algorithms

Model Comparison: Decision Tree

- Strengths of Decision Trees
  - An all-purpose classifier that does well on most problems
  - Exclude unimportant features
  - Clear rules, model can be easily interpreted
  - Fast algorithm
  - Can be used on both large and small dataset
- Weaknesses of Decision Trees
  - Model performance may suffer with complex problems
    - E.g., a large number of class labels or large number of features
  - Easy to overfit or underfit the model
  - Large trees are difficult to interpret

Model Comparison: Naïve Bayes

- Strengths of Naïve Bayes
  - Simple, fast, and very efficient
  - Do well with noise and missing data
  - Easy to obtain the predicted probability
  - Immune to overfitting: its hypothesis function is so simple it cannot accurately represent many complex situations
- Weaknesses of Naïve Bayes
  - Assumption: features are equally important and independent
  - Require to smooth for small data

Model Comparison

- Comparing Decision Tree and Naïve Bayes
  - Decision Tree with max depth = 2

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

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

target \ pred	No	Yes
No	2601	10
Yes	302	86

  - Naïve Bayes with Laplace smooth = 1

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
87.03042	90.55364	49.91763	95.01149	33.40684	92.72902	40.02642

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
86.39547	89.66511	45.49550	95.36576	26.03093	92.42762	33.11475

Compare the performances on testing set

target \ pred	No	Yes
No	2490	121
Yes	287	101

Model Comparison

- Overall model performance comparison
  - Accuracy
- Compare performances on each class
  - Precision: confidence/effectiveness of predictions
  - Recall: ability of identifying instances belonging to a class

F-measure: single metrics combines precision and recall and measures the overall performance on each class

Decision Tree Model

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

pred		
target	No	Yes
No	2601	10
Yes	302	86

TP/(TP+FP) TN/(TN+FN) TP/(TP+FN) TN/(TN+FP)

Naïve Bayes Model

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
86.39547	89.66511	45.49550	95.36576	26.03093	92.42762	33.11475

pred		
target	No	Yes
No	2490	121
Yes	287	101

Model Comparison

- Which model is better?
  - Decision Tree with max depth = 2

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

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

target \ pred	No	Yes
No	2601	10
Yes	302	86

  - Naïve Bayes with Laplace smooth = 1

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
87.03042	90.55364	49.91763	95.01149	33.40684	92.72902	40.02642

ACC	PRECISION1	PRECISION2	TPR1	TPR2	F11	F12
86.39547	89.66511	45.49550	95.36576	26.03093	92.42762	33.11475

Compare the performances on testing set

target \ pred	No	Yes
No	2490	121
Yes	287	101

$200 profit for a good car
$2,000 loss for a bad buy