Lecture 6: Business Cases and Applications

Predicting Pokemon Battle Winner

Predicting Pokemon battle winner with classification

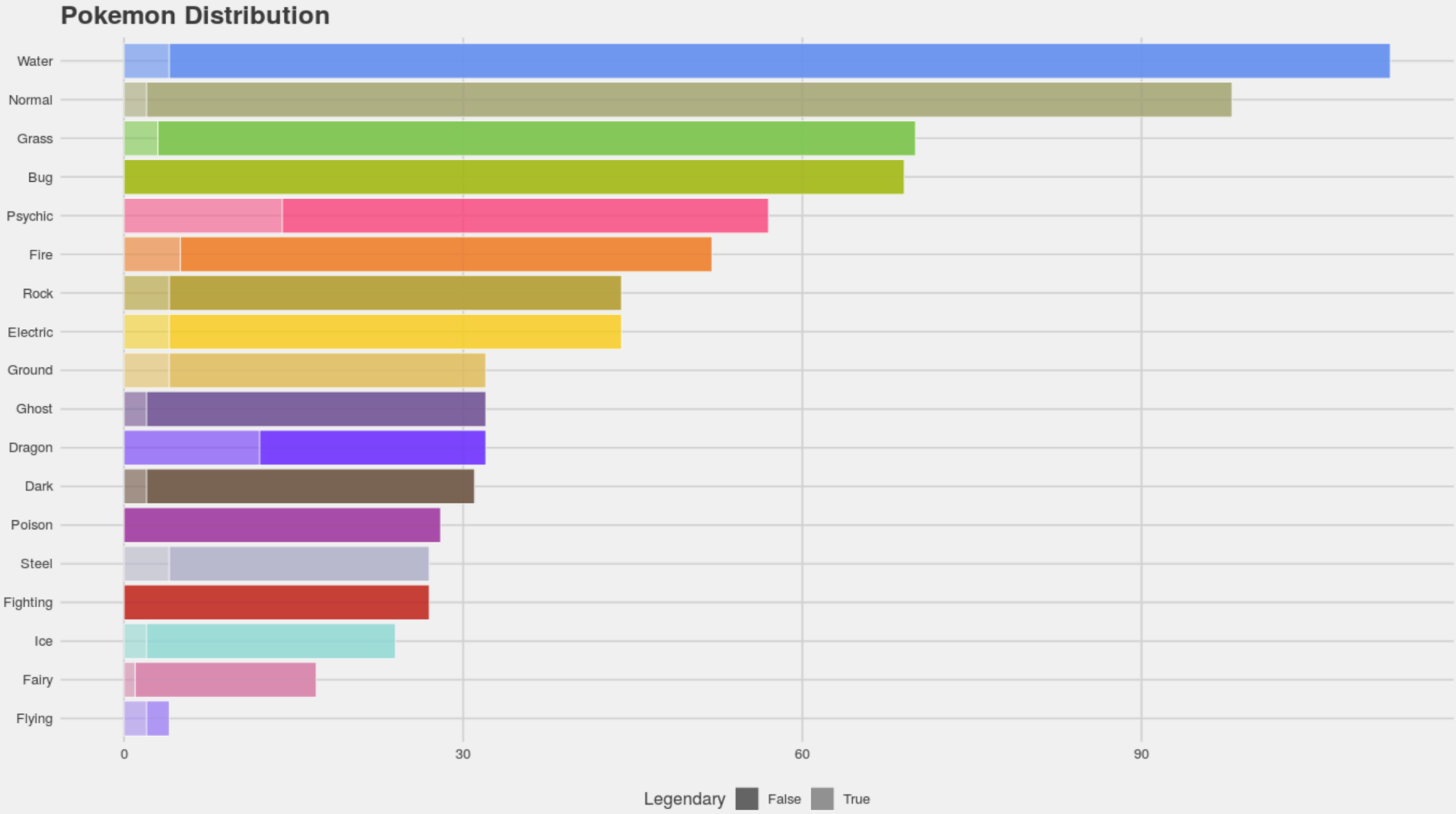
Image Source : The Verge

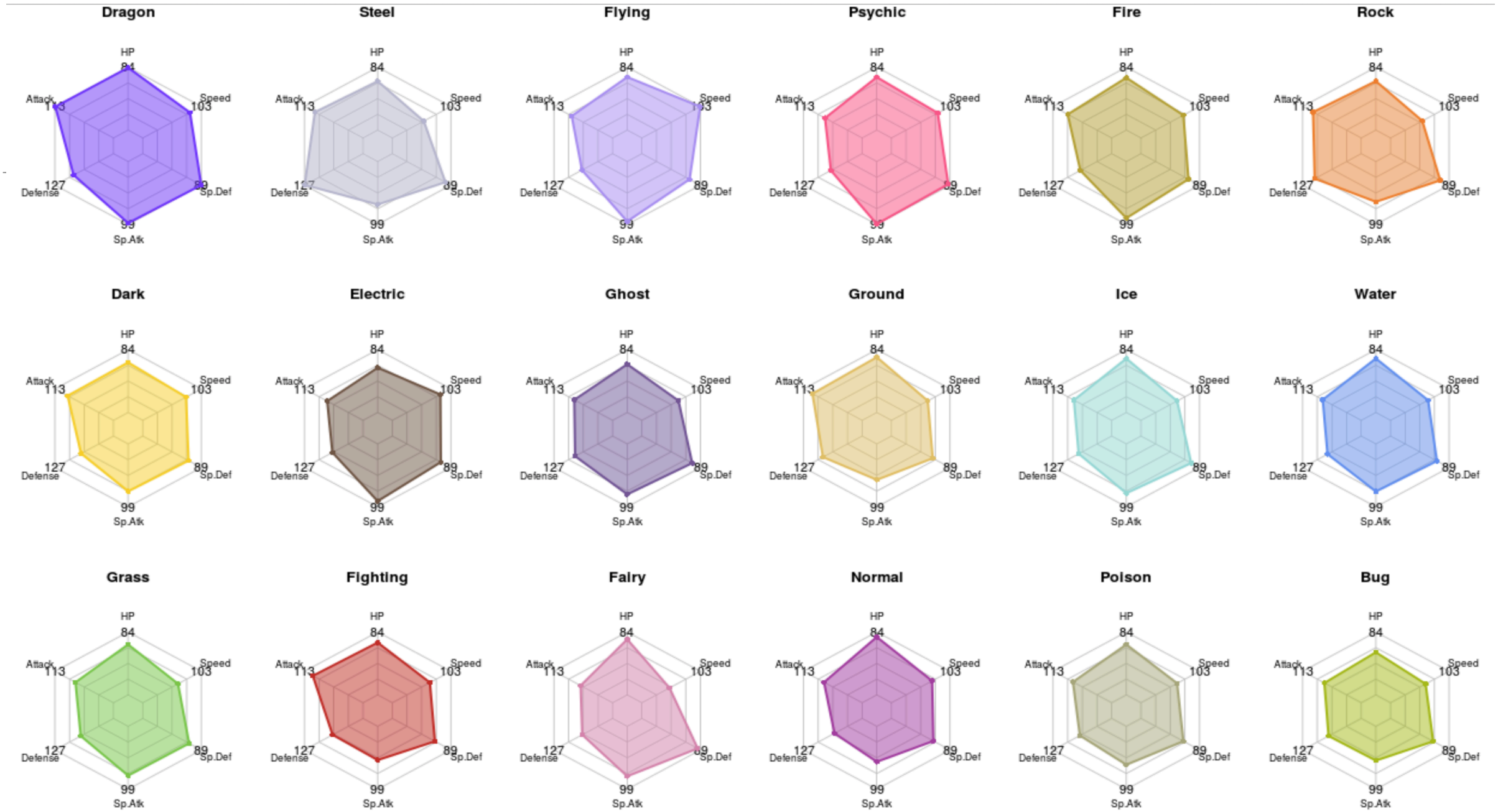
Predicting Pokemon Battle Winner

- Pokemon Information

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	FALSE
2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	FALSE
3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	FALSE
4	Mega Venus	Grass	Poison	80	100	123	122	120	80	1	FALSE
5	Charmander	Fire		39	52	43	60	50	65	1	FALSE
6	Charmeleon	Fire		58	64	58	80	65	80	1	FALSE
7	Charizard	Fire	Flying	78	84	78	109	85	100	1	FALSE
8	Mega Chariz	Fire	Dragon	78	130	111	130	85	100	1	FALSE
9	Mega Chariz	Fire	Flying	78	104	78	159	115	100	1	FALSE
10	Squirtle	Water		44	48	65	50	64	43	1	FALSE
11	Wartortle	Water		59	63	80	65	80	58	1	FALSE
12	Blastoise	Water		79	83	100	85	105	78	1	FALSE
13	Mega Blasto	Water		79	103	120	135	115	78	1	FALSE
14	Caterpie	Bug		45	30	35	20	20	45	1	FALSE
15	Metapod	Bug		50	20	55	25	25	30	1	FALSE
16	Butterfree	Bug	Flying	60	45	50	90	80	70	1	FALSE

Predicting Pokemon Battle Winner




Dragon
Steel
Flying
Psychic
Fire
Rock
Dark
Electric
Ghost
Ground
Ice
Water
Grass
Fighting
Fairy
Normal
Poison
Bug
HP
84
Speed
103
Sp.Def
89
Sp.Atk
99
Defense
127
Attack
113

Predicting Pokemon Battle Winner

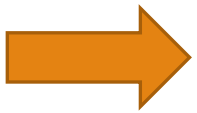
- Combat history

First_pokem	Second_poke	Winner
266	298	298
702	701	701
191	668	668
237	683	683
151	231	151
657	752	657
192	134	134
73	545	545
220	763	763
302	31	31
442	130	130
701	624	701
15	283	283
151	87	151
269	462	269
763	448	448
143	263	263
365	240	240
499	774	499

Predicting Pokemon Battle Winner

- Task: predicting battle winner
  - Construct Target variable

First_pokem	Second_poke	Winner
266	298	298
702	701	701
191	668	668
237	683	683
151	231	151
657	752	657
192	134	134
73	545	545
220	763	763
302	31	31
442	130	130
701	624	701
15	283	283
151	87	151
269	462	269
763	448	448
143	263	263
365	240	240
499	774	499



combats$Winner = ifelse(combats$Winner == combats$First_pokemon, 1, 0)

```
> combats$Winner
   [1] 0 0 0 0 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 1 0 1 0 0 1 1 0 0 0 1
  [65] 1 1 1 1 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0
 [129] 1 1 1 1 0 0 1 0 1 0 0 1 0 1 0 0 1 1 0 0 1 1 1 0 0 0 1 1 0 0 0 0 1 1
 [193] 1 0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1 0 0 1 1 0 1
 [257] 0 1 1 0 0 1 1 0 1 0 1 0 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
 [321] 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 0 0 1 1 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0
 [385] 0 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 0 1 0
 [449] 1 1 1 1 0 1 1 0 1 0 1 1 1 1 0 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0
 [513] 0 1 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0
 [577] 0 0 1 1 0 0 1 1 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 1 0 0 0 0 0 1 1 1 1 1
> summary(combats$Winner)
    0     1
26399 23601
```

Predicting Pokemon Battle Winner

- Model 1: Predicting with only combat history
- Model 2: Predicting with Pokemon Information

Predicting Pokemon Battle Winner

Predicting with only combat history

- Predictors
- Target variable

```
> str(combats)
'data.frame':   50000 obs. of  3 variables:
 $ First_pokemon : Factor w/ 784 levels "1","2","3","4",..: 257 687 183 228 144 642 184 69 212 293 ...
 $ Second_pokemon: Factor w/ 784 levels "1","2","3","4",..: 289 686 653 668 223 737 128 533 748 30 ...
 $ Winner        : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 1 ...
```

Predicting Pokemon Battle Winner

Predicting with only combat history

- Which classification method to use?
  - Decision Tree
  - Naive Bayes
  - K Nearest Neighbor

Predicting Pokemon Battle Winner

Predicting with only combat history

- Classification method
  - Decision tree
    - rpart_model <- rpart(Winner~.,data = datTrain,control = rpart.control(cp = 0.0001, maxdepth = 5))

```
> mmetric(datTrain$Winner,prediction_on_train_rpart,metric = c("ACC","PRECISION","TPR","F1"))
      ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
 91.13740   91.33871   90.90909   91.93182   90.24877   91.63430   90.57773
> mmetric(datTest$Winner,prediction_on_test_rpart,metric=c("ACC","PRECISION","TPR","F1"))
      ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
 87.73252   88.15112   87.25825   88.68544   86.66667   88.41747   86.96145
```

Which evaluation metrics are important?

Predicting Pokemon Battle Winner

Predicting with only combat history

- Classification method
  - Naïve Bayes
    - model_nb <- naiveBayes(Winner~.,data=datTrain, laplace = 1)

```
> mmetric(datTrain$Winner,prediction_on_train_rpart3,metric = c("ACC","PRECISION","TPR","F1"))
       ACC PRECISION1 PRECISION2       TPR1       TPR2         F11        F12
  87.94606   87.93828   87.95508   89.43723   86.27807   88.68142   87.10850
> mmetric(datTest$Winner,prediction_on_test_rpart3,metric=c("ACC","PRECISION","TPR","F1"))
       ACC PRECISION1 PRECISION2       TPR1       TPR2         F11        F12
  85.94573   85.91027   85.98726   87.77623   83.89831   86.83323   84.92994
```

Predicting Pokemon Battle Winner

- **Predicting with Pokemon Information**
  - Combine two datasets

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	FALSE
2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	FALSE
3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	FALSE
4	Mega Venus	Grass	Poison	80	100	123	122	120	80	1	FALSE
5	Charmander	Fire		39	52	43	60	50	65	1	FALSE
6	Charmeleon	Fire		58	64	58	80	65	80	1	FALSE
7	Charizard	Fire	Flying	78	84	78	109	85	100	1	FALSE
8	Mega Chariz	Fire	Dragon	78	130	111	130	85	100	1	FALSE
9	Mega Chariz	Fire	Flying	78	104	78	159	115	100	1	FALSE
10	Squirtle	Water		44	48	65	50	64	43	1	FALSE
11	Wartortle	Water		59	63	80	65	80	58	1	FALSE
12	Blastoise	Water		79	83	100	85	105	78	1	FALSE
13	Mega Blasto	Water		79	103	120	135	115	78	1	FALSE
14	Caterpie	Bug		45	30	35	20	20	45	1	FALSE
15	Metapod	Bug		50	20	55	25	25	30	1	FALSE
16	Butterfree	Bug	Flying	60	45	50	90	80	70	1	FALSE

First_pokem	Second_poke	Winner
266	298	298
702	701	701
191	668	668
237	683	683
151	231	151
657	752	657
192	134	134
73	545	545
220	763	763
302	31	31
442	130	130
701	624	701
15	283	283
151	87	151
269	462	269
763	448	448
143	263	263
365	240	240
499	774	499

Predicting Pokemon Battle Winner

```
'data.frame':   50000 obs. of  21 variables:
 $ first.Type.1     : Factor w/ 18 levels "Bug","Dark","Dragon",..: 16 10 5 7 16 1 15 6 1 18 ...
 $ first.Type.2     : Factor w/ 19 levels "none","Bug","Dark",..: 12 7 9 1 19 5 9 1 1 9 ...
 $ first.HP         : int  50 91 55 40 70 50 40 70 50 40 ...
 $ first.Attack     : int  64 90 40 40 60 47 50 80 65 30 ...
 $ first.Defense    : int  50 72 85 40 125 50 45 50 90 30 ...
 $ first.Sp.Atk     : int  45 90 80 70 115 57 70 35 35 55 ...
 $ first.Sp.Def     : int  50 129 105 40 70 50 45 35 35 30 ...
 $ first.Speed      : int  41 108 40 20 55 65 70 35 15 85 ...
 $ first.Generation : Factor w/ 6 levels "1","2","3","4",..: 2 5 2 2 1 5 2 1 2 3 ...
 $ first.Legendary  : Factor w/ 2 levels "False","True": 1 2 1 1 1 1 1 1 1 1 ...
 $ second.Type.1    : Factor w/ 18 levels "Bug","Dark","Dragon",..: 10 16 15 3 1 17 12 9 18 4 ..
 $ second.Type.2    : Factor w/ 19 levels "none","Bug","Dark",..: 3 7 1 1 17 10 16 4 1 1 ...
 $ second.HP        : int  70 91 75 77 20 60 65 150 50 35 ...
 $ second.Attack    : int  70 129 75 120 10 50 50 100 53 55 ...
 $ second.Defense   : int  40 90 75 90 230 150 35 120 62 40 ...
 $ second.Sp.Atk    : int  60 72 125 60 10 50 115 100 58 50 ...
 $ second.Sp.Def    : int  40 90 95 90 230 150 95 120 63 50 ...
 $ second.Speed     : int  60 108 40 48 5 60 95 90 44 90 ...
 $ second.Generation: Factor w/ 6 levels "1","2","3","4",..: 3 5 5 5 2 6 1 4 6 1 ...
 $ second.Legendary : Factor w/ 2 levels "False","True": 1 2 1 1 1 1 1 2 1 1 ...
 $ Winner           : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 1 1 1 ...
```

- **Predictors**
- **Target variable**

Predicting Pokemon Battle Winner

Predicting with Pokemon Information

- Which classification method to use?
  - Decision Tree
  - Naive Bayes
  - K Nearest Neighbor

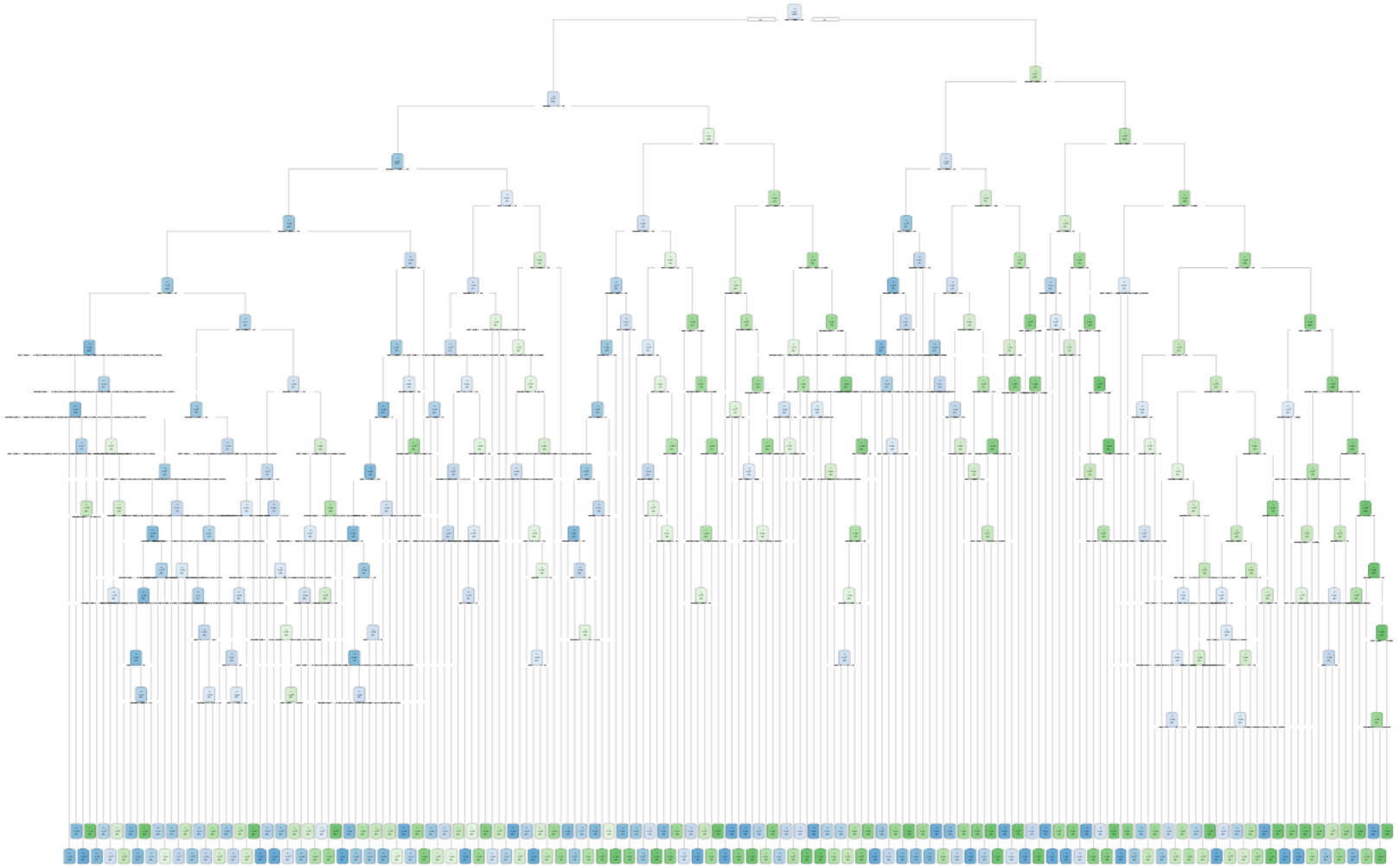
Predicting Pokemon Battle Winner

Predicting with Pokemon Information

- Classification method
  - Decision tree
  - rpart_model <- rpart(Winner~.,data = datTrain,control = rpart.control(cp = 0.0001, maxdepth = 12))

```
> mmetric(datTrain$Winner,prediction_on_train_rpart,metric = c("ACC","PRECISION","TPR","F1"))
       ACC PRECISION1 PRECISION2       TPR1       TPR2         F11        F12
  96.12297   97.27760   94.88515   95.32468   97.01592   96.29124   95.93871
> mmetric(datTest$Winner,prediction_on_test_rpart,metric=c("ACC","PRECISION","TPR","F1"))
       ACC PRECISION1 PRECISION2       TPR1       TPR2         F11        F12
  94.51297   95.97046   92.96800   93.53454   95.60734   94.73684   94.26920
```

Predicting Pokemon Battle Winner



Predicting Pokemon Battle Winner

Predicting with Pokemon Information

- Classification method
  - Naïve Bayes
  - model_nb <- naiveBayes(Winner~.,data=datTrain, laplace = 1)

```
> mmetric(datTrain$Winner,prediction_on_train_rpart3,metric = c("ACC","PRECISION","TPR","F1"))
      ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
 80.08343   81.11718   78.92537   81.17424   78.86326   81.14570   78.89431
> mmetric(datTest$Winner,prediction_on_test_rpart3,metric=c("ACC","PRECISION","TPR","F1"))
      ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
 80.19201   81.45982   78.79467   80.89405   79.40678   81.17595   79.09954
```

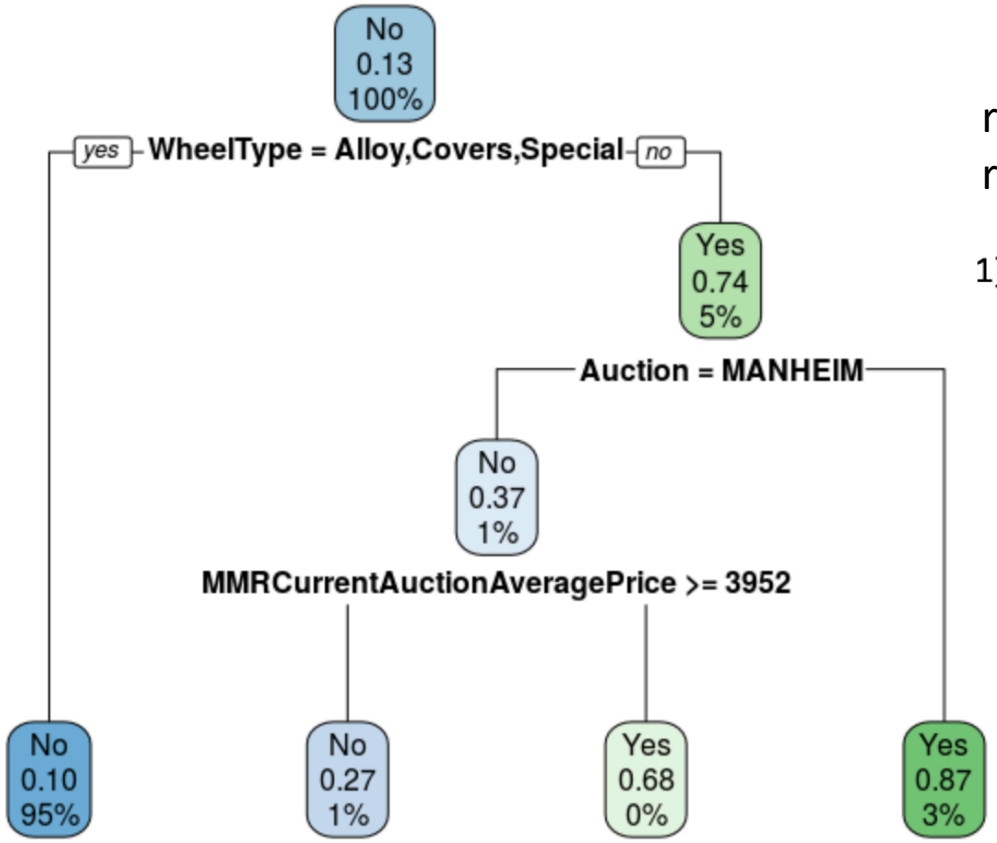
Why Naïve Bayes has worse performance?

Predicting Pokemon Battle Winner

- How we solve this problem with data mining technique?
- What are the benefits for solving this problem?
  - Which type of pokemon always win?
  - Which pokemons always win?
  - What are important predictors?
  - Why those predictors are important?

Cost-benefit Analysis for CarAuction

- Train the decision tree model on 70% carAuction data



rpart_model <- rpart(IsBadBuy~.,data = datTrain,control = rpart.control(cp = 0.0001, maxdepth = 3))

```
1) root 7001 907 No (0.87044708 0.12955292)
  2) WheelType=Alloy,Covers,Special 6674 664 No (0.90050944 0.09949056) *
  3) WheelType=unkwnWheel 327  84 Yes (0.25688073 0.74311927)
    6) Auction=MANHEIM 82  30 No (0.63414634 0.36585366)
     12) MMRCurrentAuctionAveragePrice>=3952 63  17 No (0.73015873 0.26984127) *
     13) MMRCurrentAuctionAveragePrice< 3952 19   6 Yes (0.31578947 0.68421053) *
    7) Auction=ADESA,OTHER 245  32 Yes (0.13061224 0.86938776) *
```

Cost-benefit Analysis for CarAuction

- Assume the following costs:
  - Profit when we resell a good car: $600
  - Loss when we buy a bad car: $5000
  - Opportunity cost of passing up a good car: $600

```
> mmetric(datTrain$IsBadBuy,prediction_on_train,metric=c("ACC","PRECISION","TPR","F1"))
       ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
  89.73004   89.89164   85.60606   99.37644   24.91731   94.39638   38.59949
> mmetric(datTest$IsBadBuy,prediction_on_test,metric=c("ACC","PRECISION","TPR","F1"))
       ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
  89.59653   89.67898   87.25490   99.50211   22.93814   94.33551   36.32653
```

	pred	
target	No	Yes
No	2598	13
Yes	299	89

Cost-benefit Analysis for CarAuction

- Benefit= 2598 * 600 = 1,558,800
- Loss = 299 * 5000 = 1,495,000
- Opportunity cost = 600* 13 = 7,800

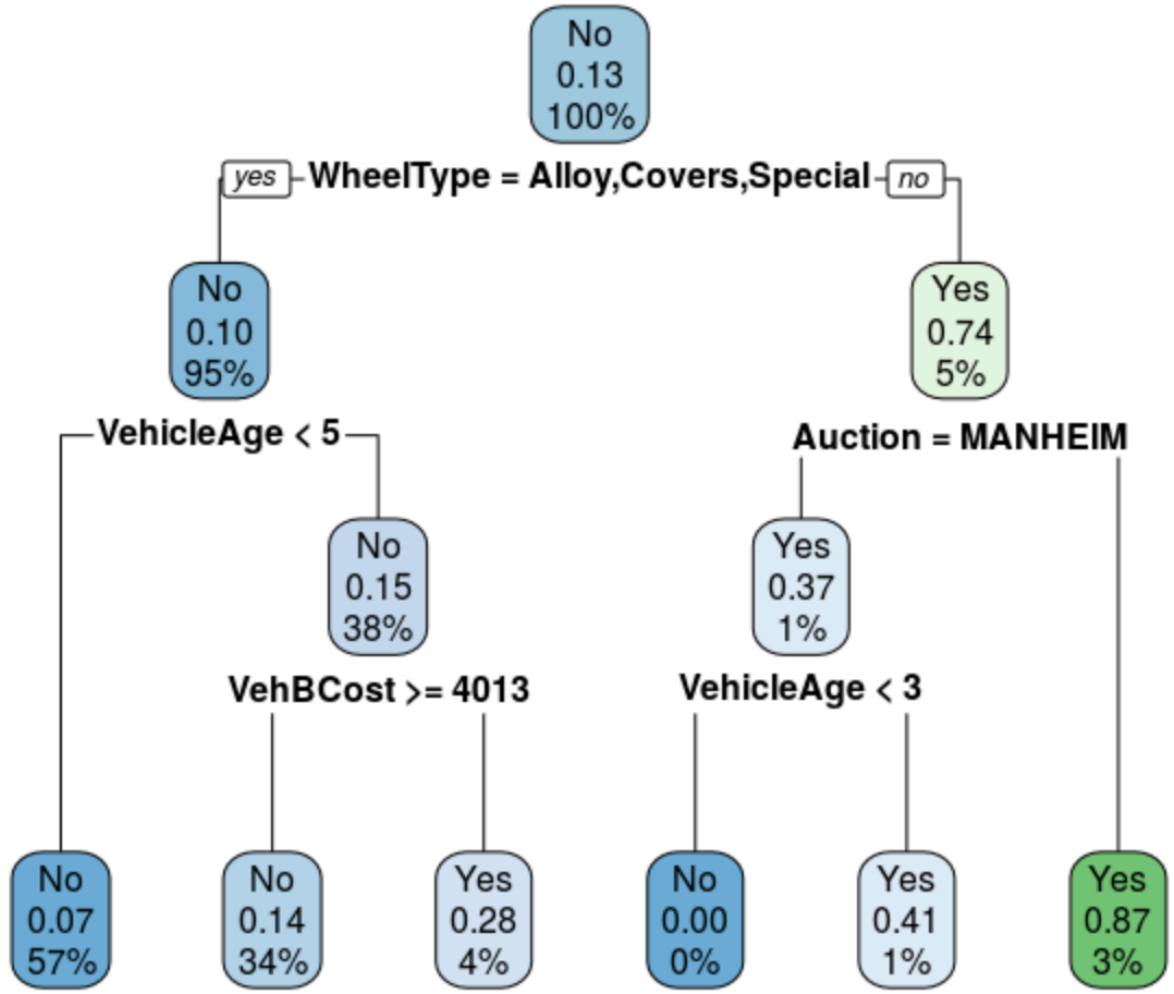
- Benefit – loss – opportunity cost = 56,000

Cost-benefit Analysis for CarAuction

- Setting cost matrix

```
> print(costMatrix)
    predict_No Predict_Yes
No           0           1
Yes          5           0
```

rpart_model2 <- rpart(IsBadBuy~.,data = datTrain,control = rpart.control(cp = 0.0001, maxdepth = 3), parms = list(loss = costMatrix))



Cost-benefit Analysis for CarAuction

- Evaluation results with cost

```
> mmetric(datTrain$IsBadBuy,prediction_on_train2,metric=c("ACC","PRECISION","TPR","F1"))
       ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
  87.71604   90.85233   53.94958   95.50377   35.39140   93.12000   42.74301
> mmetric(datTest$IsBadBuy,prediction_on_test2,metric=c("ACC","PRECISION","TPR","F1"))
       ACC PRECISION1 PRECISION2       TPR1       TPR2        F11        F12
  87.92931   90.26137   56.31068   96.55304   29.89691   93.30126   39.05724
```

	pred	
target	No	Yes
No	2521	90
Yes	272	116

Cost-benefit Analysis for CarAuction

- Benefit= 2521 * 600 = 1,512,600
- Loss = 272 * 5000 = 1,360,000
- Opportunity cost = 600* 90 = 54,000

- Benefit – loss – opportunity cost = 98,600

Application

- You work for a law firm that offers various legal services to both individuals and organizations, including **contracts, accident litigation, medical malpractice and settlement, work-related injury and compensation, and intellectual property**. Individual clients contribute to more than 80% of the firm's revenues; Marketing Department decides to do radio advertising to attract more individual clients and prospective customers. For each type of legal services common sought by individual clients, Marketing Department has produced a focused advertisement and wants to air the advertisement through appropriate radio stations.

Application

- There are many radio stations in the Greater Salt Lake area that noticeably differ in content/music and targeted listeners; Marketing manager wants to focus on the **10 most popular radio stations**, each of which can provide the firm with a typical profile of its listeners. Marketing manager wants to **advertise each legal service** with **two radio stations** most effective for attracting listeners who frequently need or seek this particular legal service. Marketing manager is now at your door, asking you how to select which two radio stations to air the advertisement for each legal service.

Firm Customer Dataset

customerID	name	DOB	gender	mailing address	contact person	phone numbers	education backgroud	marital status	job type	income	LegalServiceRequested
10157	Dangelo Roberts	7/1/77	male	78 Luna Street Moroni, UT 84646	Clay Mullen	(838) 019-1185	high school diploma	single	blue collar	56806	work injury
10329	Lyla Harrell	10/24/72	male	9017 Court Street Dammeron Valley, UT 84783	Adrian Osborne	(943) 663-3279	master degree	married	technician	1E+05	intellectual property
10385	Addyson Cervantes	2/8/39	male	26 Nichols Dr. Rush Valley, UT 84069	Gabriela Mason	(700) 338-6226	bachelor degree	married	retired	86103	medical settlement
10584	Ashlynn Armstrong	5/13/78	female	650 E. Frost Dr. Lindon, UT 84042	Brylee Morgan	(324) 892-3430	bachelor degree	single	management	37636	work injury
10839	Jude Perez	11/10/45	male	7 Hilldale Drive Snowville, UT 84336	Tucker Kirby	(761) 747-5772	bachelor degree	divorced	retired	79816	medical settlement
11020	Tyree Robles	1/15/36	female	61 Country Club St. Salt Lake City, UT 84150	Jake Carpenter	(956) 156-8544	bachelor degree	divorced	retired	82159	medical settlement
11321	Aiden Pearson	12/12/75	male	24 Law Lane Salt Lake City, UT 84129	Nolan Parrish	(731) 225-3802	master degree	divorced	technician	1E+05	intellectual property
11639	Malachi Morales	9/26/38	female	95 Fletcher St. Herriman, UT 84096	Alessandra Bernard	(793) 322-5663	bachelor degree	single	technician	66883	medical settlement
12089	Dwayne Garcia	3/23/82	female	7940 10th Street Delta, UT 84624	Felipe Yu	(200) 255-0054	bachelor degree	divorced	management	49404	work injury
12682	Raymond Rogers	9/13/75	male	9373 West Gartner Ave. Corinne, UT 84307	Lilyana Rice	(512) 287-2706	master degree	married	technician	1E+05	intellectual property
12787	Jaiden Burke	11/19/91	male	2 East Frost Court Cornish, UT 84308	Andy Graves	(120) 851-5091	bachelor degree	married	sales	62994	Contracts
12830	Chanel Henderson	2/11/91	female	35 Windfall Road Payson, UT 84651	Javion Villanueva	(695) 460-7557	master degree	divorced	admin	39640	Contracts
12845	Colton Chandler	6/1/88	female	327 Jade Lane Clawson, UT 84516	Cedric Houston	(497) 138-5364	master degree	divorced	admin	45435	Contracts
12847	Karla Bishop	7/5/85	male	1 Golden Star St. Alton, UT 84710	Gwendolyn Dickson	(430) 421-3681	bachelor degree	married	management	56214	Contracts
13262	Landyn Mccarty	9/28/78	female	7934 W. High Point Ave. Santaquin, UT 84655	Kaiden Thomas	(231) 729-3425	bachelor degree	single	blue collar	91279	intellectual property
13302	Cheyanne Dominguez	10/6/95	female	25 Trusel Drive Bicknell, UT 84715	Isaias Montoya	(801) 278-5735	bachelor degree	married	student	26355	accident litigation
13452	Kennedy Brennan	8/3/92	female	695 Hill Field Drive Salt Lake City, UT 84147	Viviana Berger	(250) 017-7140	master degree	single	admin	46156	Contracts
13457	Roger Carter	3/12/00	male	9920 Noble St. Ogden, UT 84415	Violet Gregory	(677) 507-9234	high school diploma	single	blue collar	28824	accident litigation
13682	Russell Burch	7/19/84	female	7710 Oakland Rd. Dutch John, UT 84023	Kassandra Blackburn	(558) 000-8636	bachelor degree	divorced	management	47111	work injury
13728	Evan Sweeney	10/9/81	female	65 Bath Court Elberta, UT 84626	Ronnie Walsh	(668) 898-7945	doctoral degree	divorced	expert	73506	intellectual property
13752	Trinity Buck	8/14/96	female	8973 Crystal Street Altonah, UT 84002	Carolina Webb	(650) 769-6746	bachelor degree	divorced	student	21038	accident litigation
13810	Leonidas Holden	2/27/36	female	77 Art Rd. Ogden, UT 84415	Tori Hampton	(710) 913-2350	bachelor degree	single	retired	71898	medical settlement
13838	Gage Moss	5/20/81	male	215 Sunshine Ave. Tropic, UT 84776	Monique Blevins	(553) 706-6658	high school diploma	divorced	blue collar	45337	work injury
14107	Osvaldo Stuart	9/5/76	male	995 Penrose St. Provo, UT 84606	Jasmine Schmitt	(762) 437-2769	high school diploma	divorced	blue collar	59860	work injury
14207	Brenden Garrett	9/28/79	male	882 Hart Street Salt Lake City, UT 84128	Vicente Ashley	(817) 383-0977	high school diploma	single	blue collar	52631	work injury
14213	Hayden Mays	4/30/43	male	93 Trinity St. Moab, UT 84532	Alisha Fletcher	(314) 501-5846	bachelor degree	divorced	retired	75142	medical settlement
14526	Franklin Schaefer	8/30/82	female	87 Archer Street Honeyville, UT 84314	Trenton Zuniga	(352) 542-0287	master degree	divorced	admin	46215	Contracts
14606	Deangelo Pollard	8/10/72	male	9691 Elmwood Ave. Logan, UT 84322	Ally Mckee	(131) 971-2737	master degree	married	technician	1E+05	intellectual property
14680	Ashanti Dillon	2/4/76	female	7774 Temple Street Salt Lake City, UT 84134	Jadyn Walton	(578) 686-6252	bachelor degree	single	management	49937	work injury
14731	Ayaan Johnston	5/2/42	female	675 Sutor St. Bicknell, UT 84715	Darrell Snow	(415) 967-4435	bachelor degree	married	retired	68352	medical settlement

Radio Listener Profile

Radio Statio	Gender	age	education ba	income	job type	lifestyle
classic rock	male	55	doctoral deg	27k	blue collar	hippie
politics	male	44	master degr	120k	technician	Simple living
classic 60s/7	male	75	bachelor deg	81k	retired	Empty nesters
top 40	female	23	bachelor deg	25k	student	activism
pop music	female	38	college degr	44k	managemen	Quirkyalone
religion	female	38	doctoral deg	98k	expert	Asceticism
soft music	female	80	bachelor deg	71k	retired	Vegetarianism
jazz	female	30	master degr	44k	admin	Rural lifestyle
talk shows	male	45	bachelor deg	55k	sales	traditional
sports	male	25	high school c	54k	blue collar	Communality living

Goal

- Identifying two radio stations for each legal service type

Legal Service Type	Radio Station
work injury	???
	???
medical settlement	???
	???
intellectual property	???
	???
Contracts	???
	???
accident litigation	???
	???

classic rock
politics
classic 60s/7
top 40
pop music
religion
soft music
jazz
talk shows
sports

```
> str(LawFirm)
'data.frame':   500 obs. of  12 variables:
 $ customerID           : int  10157 10329 10385 10584 10839 11020 11321 11639 12089 12682 ...
 $ name                 : Factor w/ 500 levels "Abbie Joyce",..: 133 347 6 59 266 484 14 353 154 421 ...
 $ DOB                  : Factor w/ 486 levels "1/1/1935","1/10/1937",..: 365 63 199 292 89 6 134 469 226 455 ...
 $ gender               : Factor w/ 2 levels "female","male": 2 2 2 1 2 1 2 1 1 2 ...
 $ mailing.address      : Factor w/ 500 levels "1 Bridge Street Brian Head, UT 84719",..: 313 422 80 230 248 213 70 463 328 447 ...
 $ contact.person       : Factor w/ 500 levels "Aaron Mcgee",..: 110 4 178 90 466 225 384 13 173 313 ...
 $ phone.numbers        : Factor w/ 500 levels "(100) 008-6966",..: 429 477 346 144 380 482 358 397 76 240 ...
 $ education.backgroud  : Factor w/ 4 levels "bachelor degree",..: 3 4 1 1 1 1 4 1 1 4 ...
 $ marital.status       : Factor w/ 3 levels "divorced","married",..: 3 2 2 3 1 1 1 3 1 2 ...
 $ job.type             : Factor w/ 8 levels "admin","blue collar",..: 2 8 5 4 5 5 8 8 4 8 ...
 $ income               : int  56806 101099 86103 37636 79816 82159 108978 66883 49404 109813 ...
 $ LegalServiceRequested: Factor w/ 5 levels "accident litigation",..: 5 3 4 5 4 4 3 4 5 3 ...
> str(Radio)
'data.frame':   10 obs. of  7 variables:
 $ Radio.Station       : Factor w/ 10 levels "classic 60s/70s",..: 2 4 1 10 5 6 7 3 9 8
 $ Gender              : Factor w/ 2 levels "female","male": 2 2 2 1 1 1 1 1 2 2
 $ age                 : int  55 44 75 23 38 38 80 30 45 25
 $ education.background: Factor w/ 5 levels "bachelor degree",..: 3 5 1 1 2 3 1 5 1 4
 $ income              : Factor w/ 9 levels "120k","25k","27k",..: 3 1 8 2 4 9 7 4 6 5
 $ job.type            : Factor w/ 8 levels "admin","blue collar",..: 2 8 5 7 4 3 5 1 6 2
 $ lifestyle           : Factor w/ 10 levels "activism","Asceticism",..: 5 8 4 1 6 2 10 7 9 3
```

Data Transformation

```
> str(LawFirm)
'data.frame':   500 obs. of  6 variables:
 $ gender               : Factor w/ 2 levels "female","male": 2 2 2 1 2 1 2 1 1 2 ...
 $ education.background : Factor w/ 5 levels "bachelor degree",..: 3 4 1 1 1 1 4 1 1 4 ...
 $ job.type             : Factor w/ 8 levels "admin","blue collar",..: 2 8 5 4 5 5 8 8 4 8 ...
 $ income               : int  56806 101099 86103 37636 79816 82159 108978 66883 49404 109813 ...
 $ LegalServiceRequested: Factor w/ 5 levels "accident litigation",..: 5 3 4 5 4 4 3 4 5 3 ...
 $ age                  : num  42.2 46.9 80.6 41.4 73.9 ...
> str(Radio)
'data.frame':   10 obs. of  5 variables:
 $ gender              : Factor w/ 2 levels "female","male": 2 2 2 1 1 1 1 1 2 2
 $ age                 : int  55 44 75 23 38 38 80 30 45 25
 $ education.background: Factor w/ 5 levels "bachelor degree",..: 3 5 1 1 2 3 1 5 1 4
 $ income              : num  27000 120000 81000 25000 44000 98000 71000 44000 55000 54000
 $ job.type            : Factor w/ 8 levels "admin","blue collar",..: 2 8 5 7 4 3 5 1 6 2
```

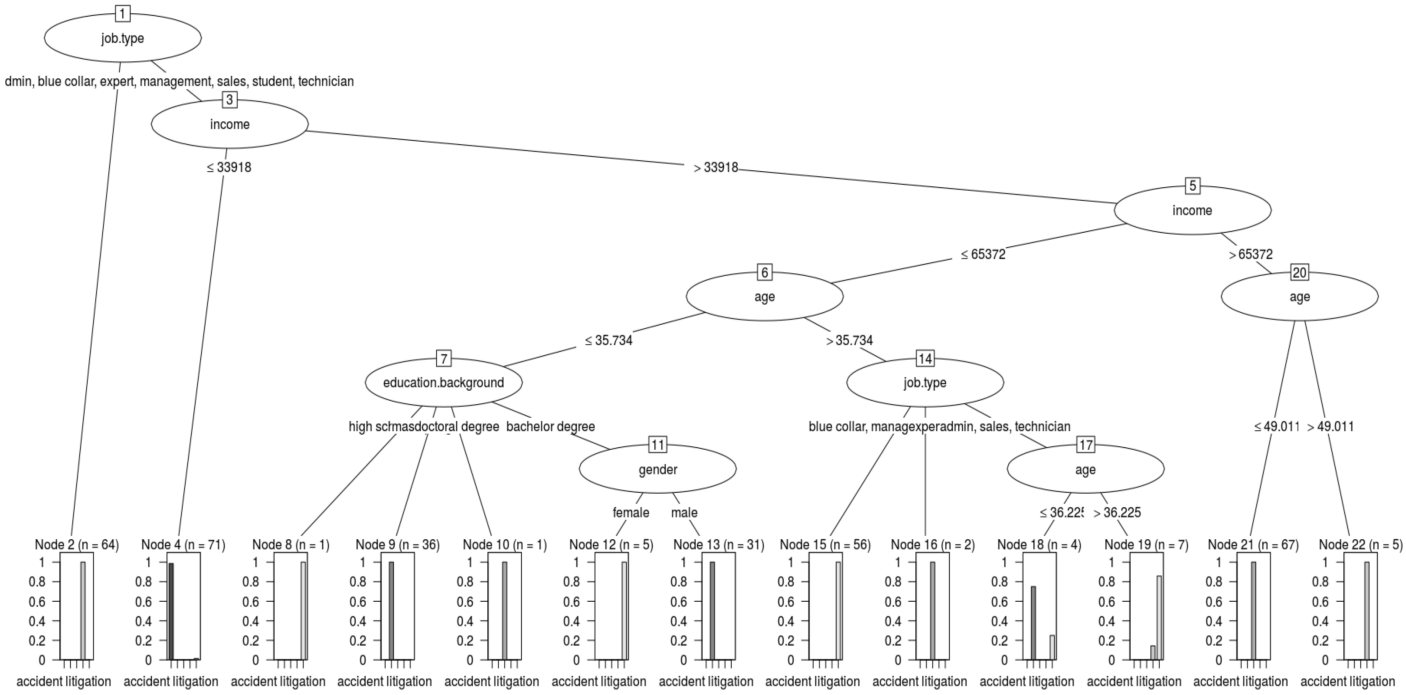
Classification

- Predictors
  - Gender, education background, job type, income, age

- Target variable
  - Legal service type

Classification

- Model training and evaluation
  - Split the Law Firm data, evaluate model performance on training and testing data
    - model=C5.0(LegalServiceRequested ~ ., data=datTrain, control = C5.0Control(CF = 0.5))



Classification

- Confusion matrix

```
> cm_train
                      Predicted
Actual                 accident litigation Contracts intellectual property medical settlement work injury
  accident litigation                   70         0                     0                 0           0
  Contracts                              0        70                     0                 0           0
  intellectual property                  0         0                    70                 0           0
  medical settlement                     0         0                     0                69           1
  work injury                            1         1                     0                 0          68
> cm_test
                      Predicted
Actual                 accident litigation Contracts intellectual property medical settlement work injury
  accident litigation                   30         0                     0                 0           0
  Contracts                              0        27                     1                 0           2
  intellectual property                  0         0                    29                 1           0
  medical settlement                     0         0                     0                30           0
  work injury                            0         0                     0                 0          30
```

Classification

- Accuracy, prediction, recall, and F-measure

```
[1] 0.9914286
                       precision    recall        f1
 accident litigation   0.9859155 1.0000000 0.9929078
 Contracts             0.9859155 1.0000000 0.9929078
 intellectual property 1.0000000 1.0000000 1.0000000
 medical settlement    1.0000000 0.9857143 0.9928058
 work injury           0.9855072 0.9714286 0.9784173
```

Training performance

```
[1] 0.9733333
                       precision    recall        f1
accident litigation    1.0000000 1.0000000 1.0000000
Contracts              1.0000000 0.9000000 0.9473684
intellectual property  0.9666667 0.9666667 0.9666667
medical settlement     0.9677419 1.0000000 0.9836066
work injury            0.9375000 1.0000000 0.9677419
```

Testing performance

Classification

- Train the decision tree model on the whole LawFirm data
  - model_LawFirm=C5.0(LegalServiceRequested ~ ., data=LawFirm, control = C5.0Control(CF = 0.5))
- Predict legal service type for radio listeners profile
  - predicted_radios = predict(model_LawFirm, newdata = Radio, type = "class")

Classification

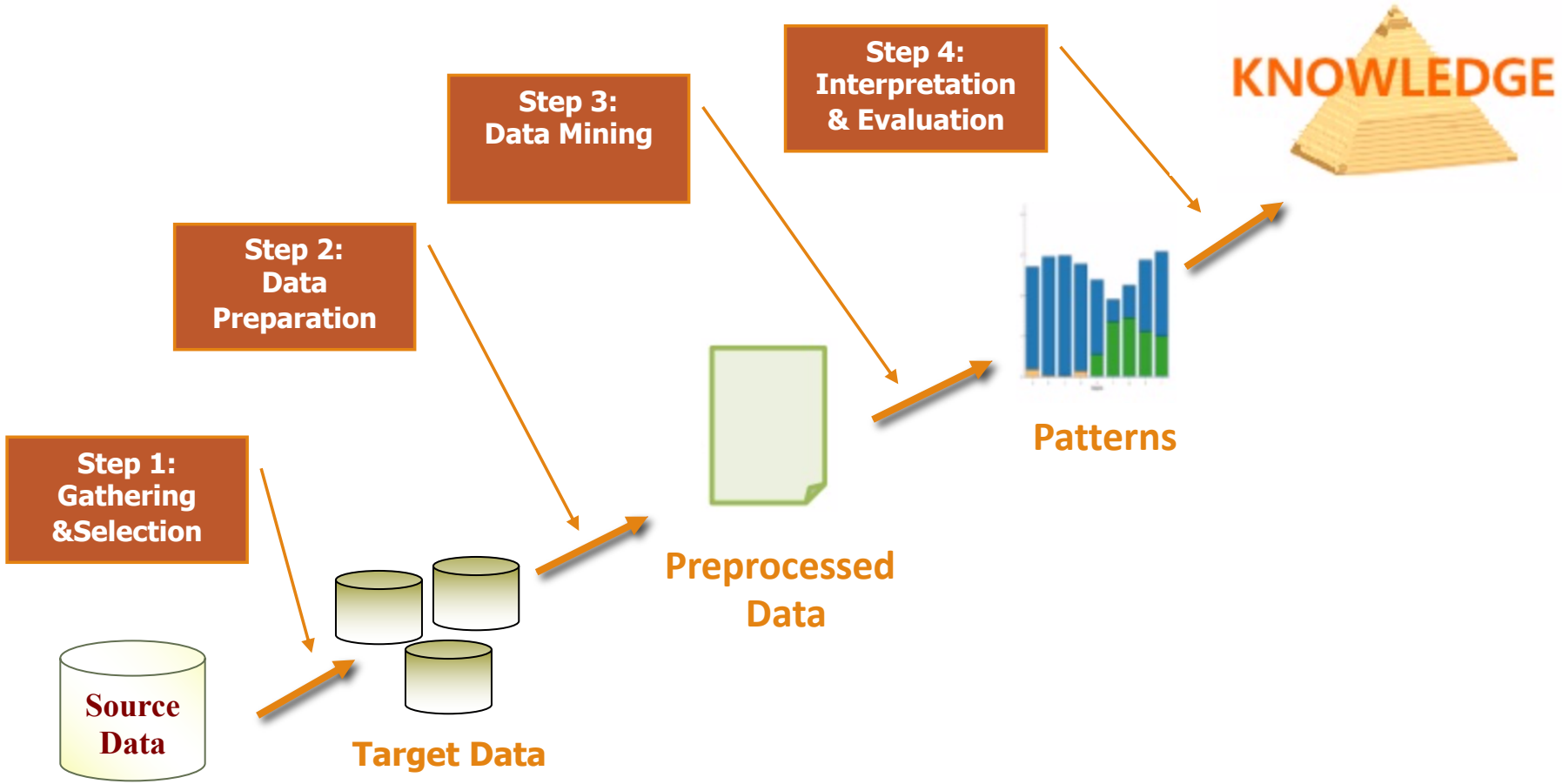
	Radio.Station	Gender	age	education.background	income	job.type	lifestyle	radio
1	classic rock	male	55	doctoral degree	27k	blue collar	hippie	accident litigation
2	politics	male	44	master degree	120k	technician	Simple living	intellectual property
3	classic 60s/70s	male	75	bachelor degree	81k	retired	Empty nesters	medical settlement
4	top 40	female	23	bachelor degree	25k	student	activism	accident litigation
5	pop music	female	38	college degree	44k	management	Quirkyalone	work injury
6	religion	female	38	doctoral degree	98k	expert	Asceticism	intellectual property
7	soft music	female	80	bachelor degree	71k	retired	Vegetarianism	medical settlement
8	jazz	female	30	master degree	44k	admin	Rural lifestyle	Contracts
9	talk shows	male	45	bachelor degree	55k	sales	traditional	work injury
10	sports	male	25	high school diploma	54k	blue collar	Communality living	work injury

Classification

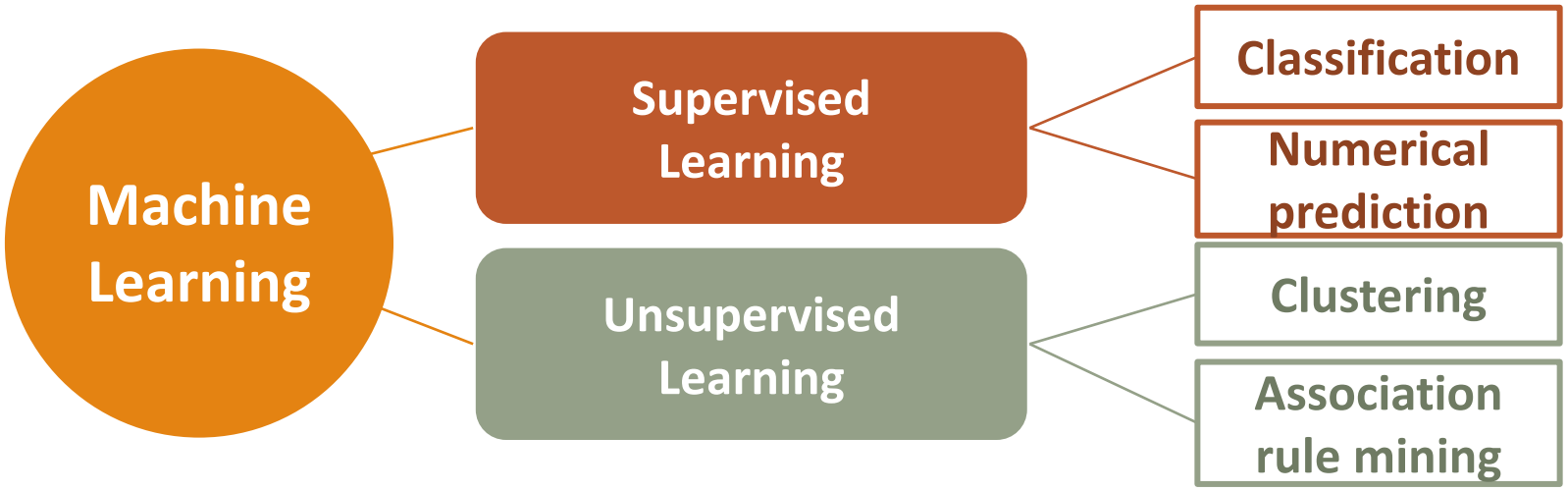
```
> predicted_radios = predict(model_LawFirm, newdata = Radio, type = "prob")
> predicted_radios
   accident litigation   Contracts intellectual property medical settlement work injury
1         0.98235294 0.001960784           0.001960784        0.001960784 0.011764706
2         0.00204081 0.002040816           0.991836735        0.002040816 0.002040816
3         0.00219780 0.002197802           0.002197802        0.991208791 0.002197802
4         0.98235294 0.001960784           0.001960784        0.001960784 0.011764706
5         0.00240963 0.002409639           0.002409639        0.002409639 0.990361446
6         0.00204081 0.002040816           0.991836735        0.002040816 0.002040816
7         0.00219780 0.002197802           0.002197802        0.991208791 0.002197802
8         0.00222222 0.991111111           0.002222222        0.002222222 0.002222222
9         0.02500000 0.025000000           0.025000000        0.025000000 0.899999999
10        0.04000000 0.040000000           0.240000000        0.040000000 0.639999999
```

Exam Review

Data Mining Process



Data Mining Tasks (multiple choice)



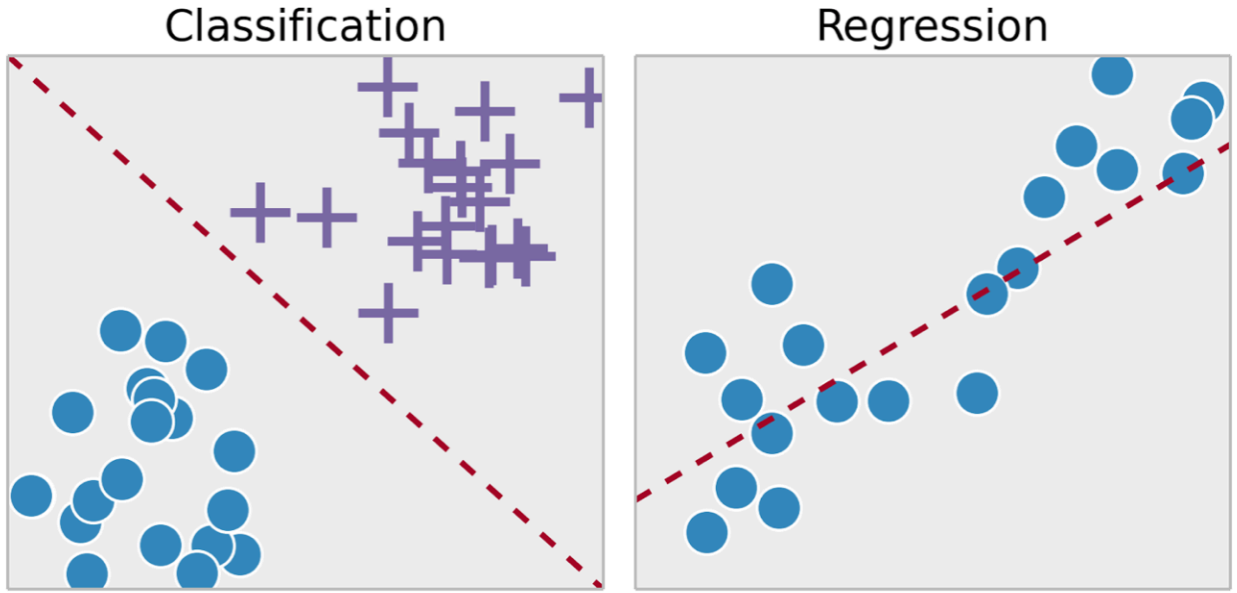
Data Mining Tasks

- **Supervised learning- Classification**
  - Predicting Pokemon battle winner with classification

Image Source : The Verge

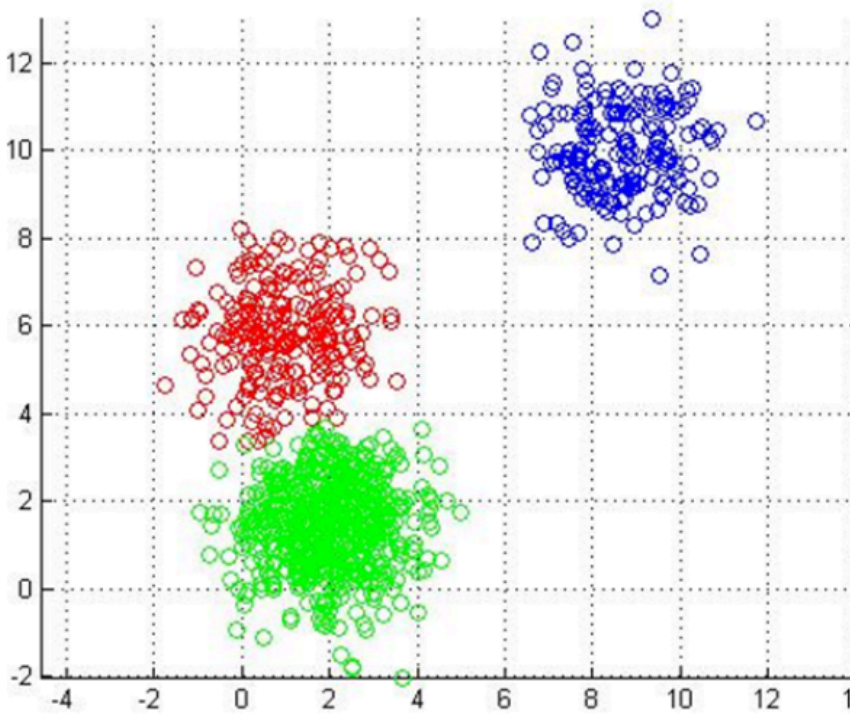
Data Mining Tasks

- **Supervised learning-Numerical Prediction**
  - A numerical prediction problem is when the output variable is a real value, such as “dollars” or “weight”.



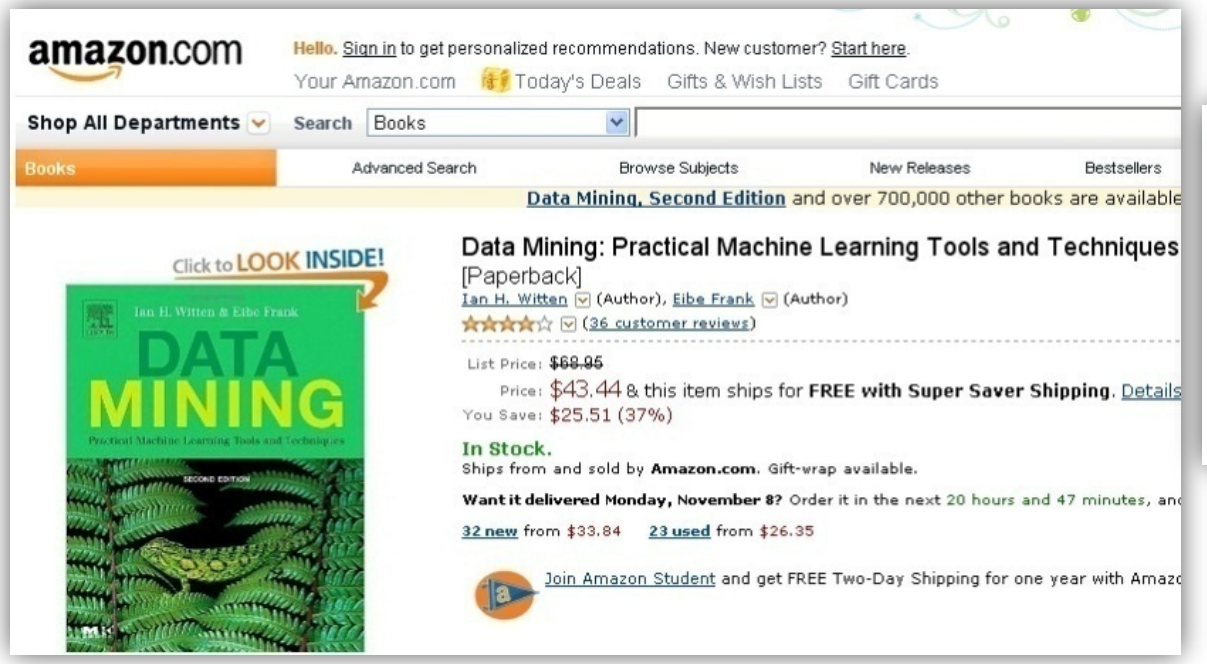
Data Mining Tasks

- **Unsupervised learning-Clustering**
  - Customer segmentation
    - Retailers want to segment customers based on their spend patterns and understand price sensitivity of the customers. Some of the segmentation variables considered are – total spend, value of discounts, number of items bought on discounts.



Data Mining Tasks

- **Unsupervised learning-Association rule mining**



Promotions to
retain customers
& increase sales

Data Exploration (multiple choice)

- **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

Data Exploration

- **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

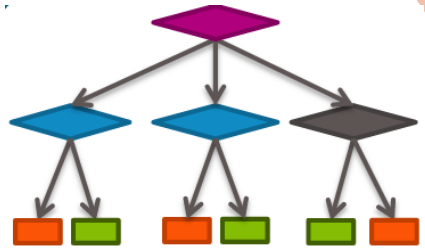
Data Exploration

- The better you understand your data, the better you will be able to match a machine learning model to your learning problem.
  - Correlation
  - Outliers
  - Missing values
  - Variable distribution

Decision Tree (multiple choice + short answer)

Information of Cars

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



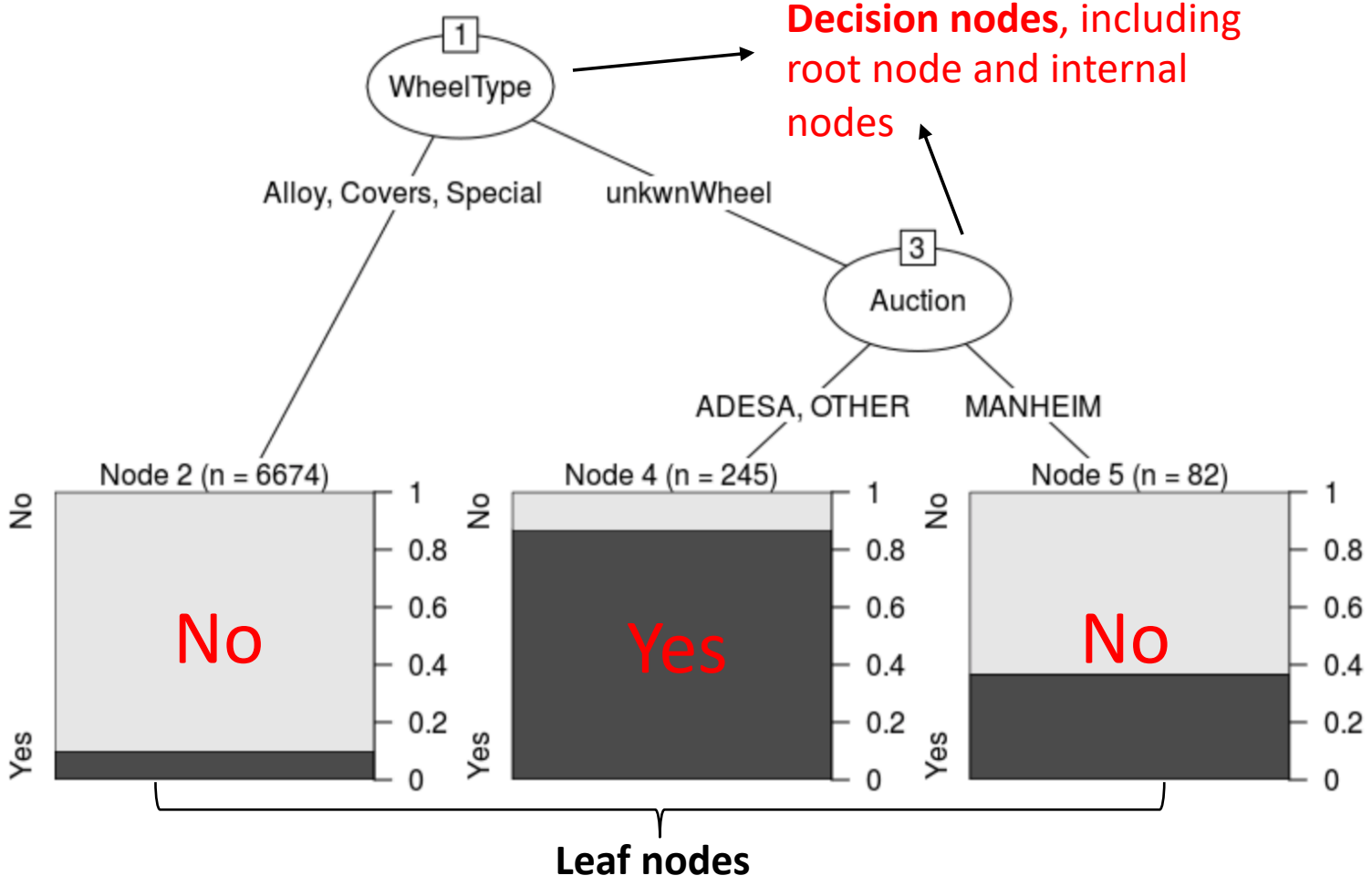
IsBadBuy

Predictors (X)

Classifier

Target variable /Label (Y)

Decision Tree

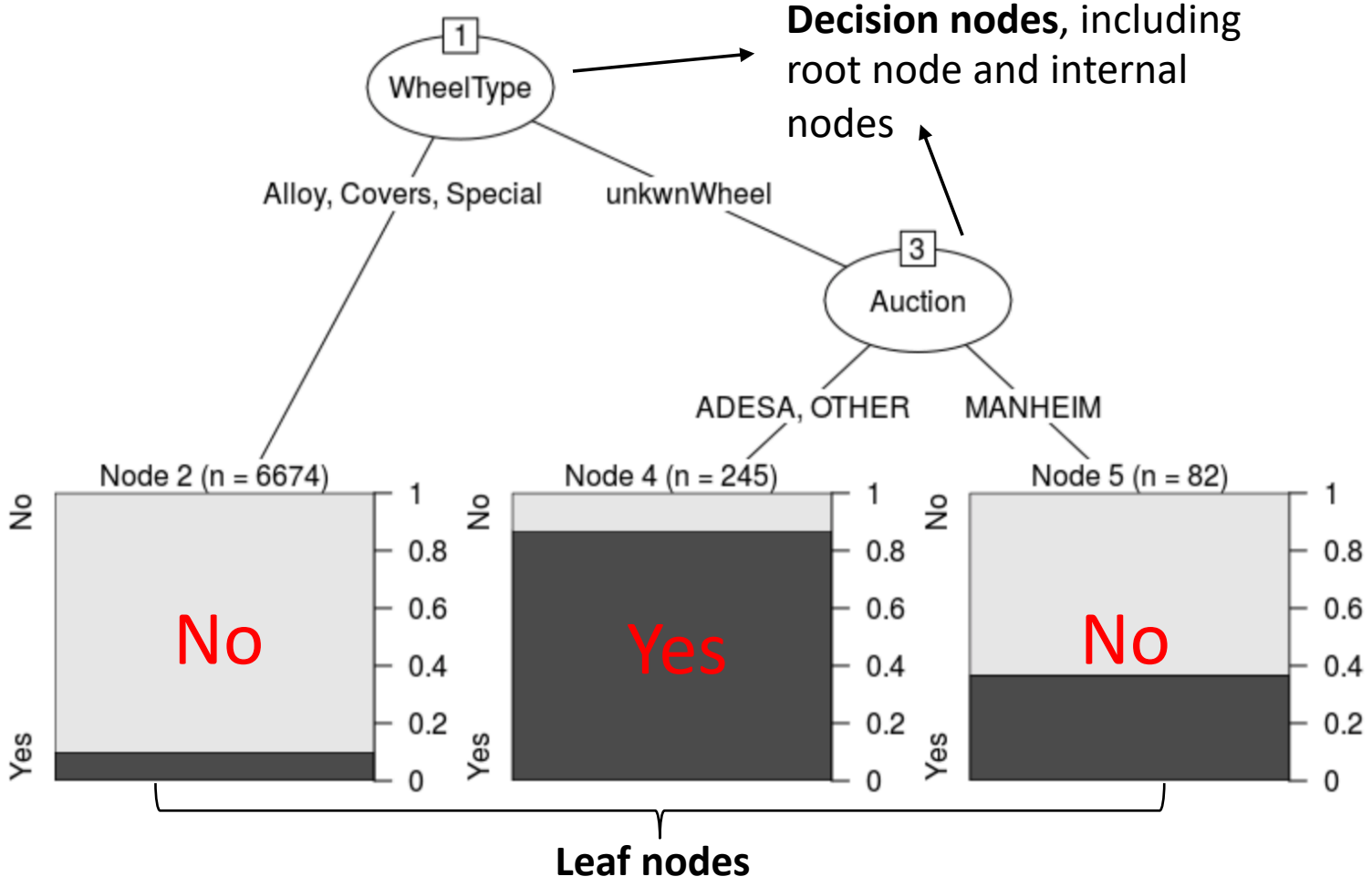


Decision nodes, including root node and internal nodes

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



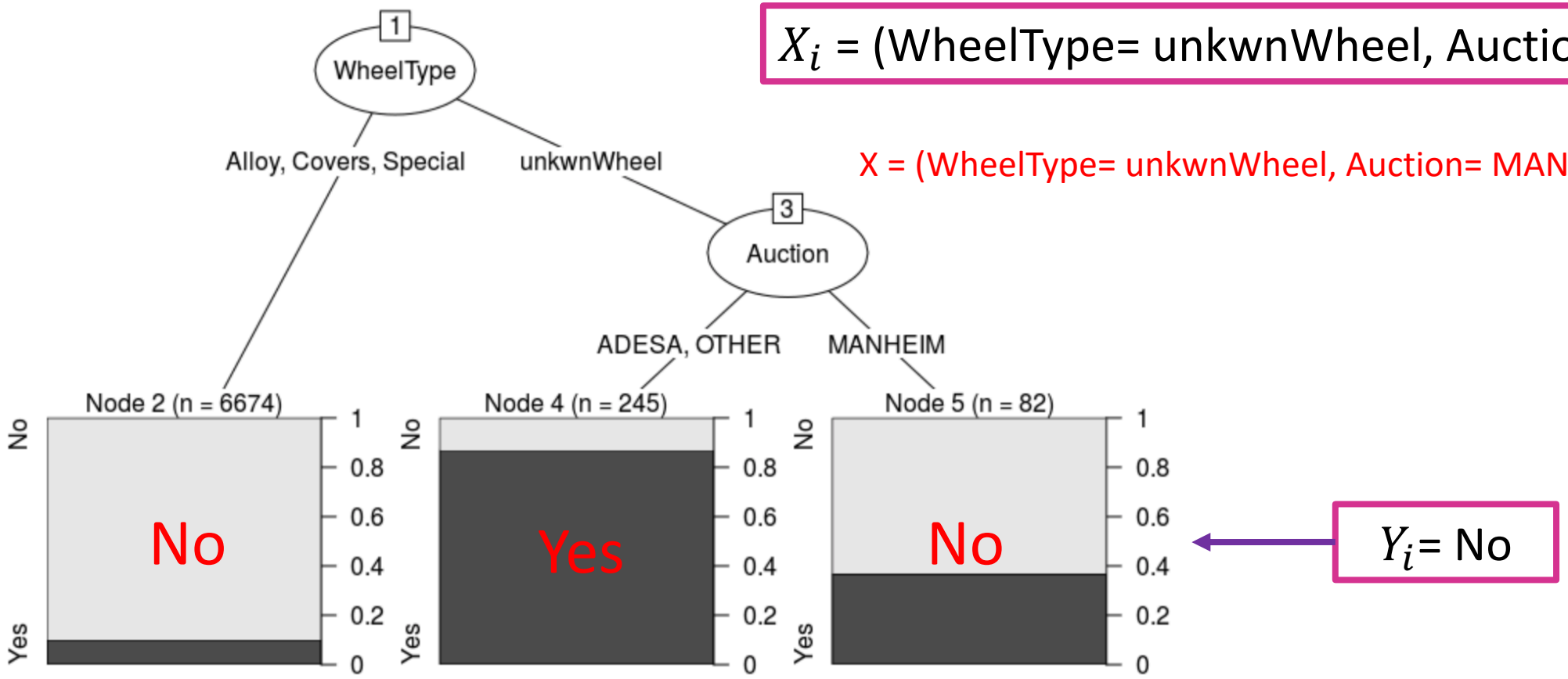
Decision nodes, including root node and internal nodes

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



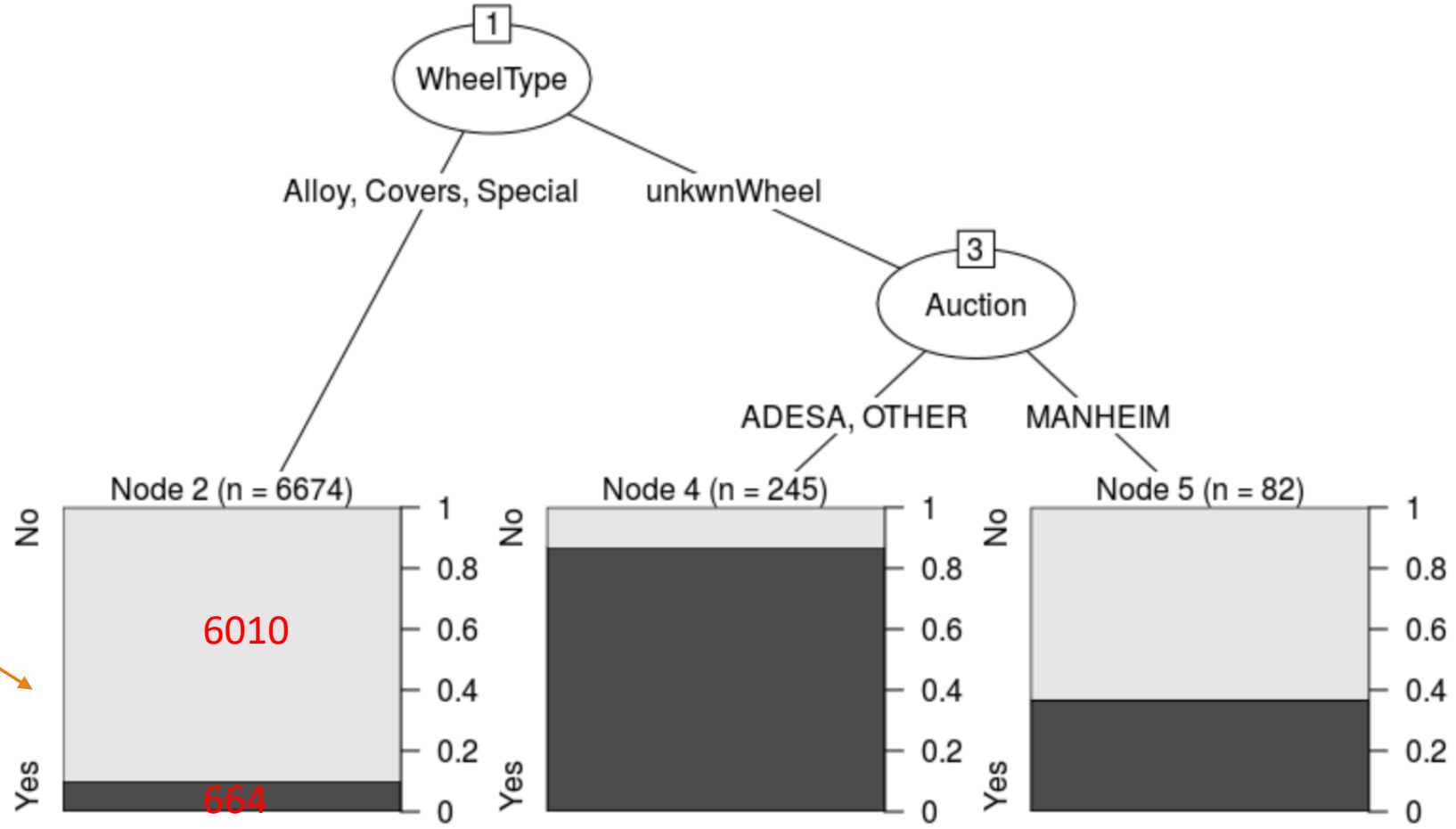
X_i = (WheelType= unkwnWheel, Auction= MANHEIM)

X = (WheelType= unkwnWheel, Auction= MANHEIM, Color = Red)?

Y_i= No

Decision Tree

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



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

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

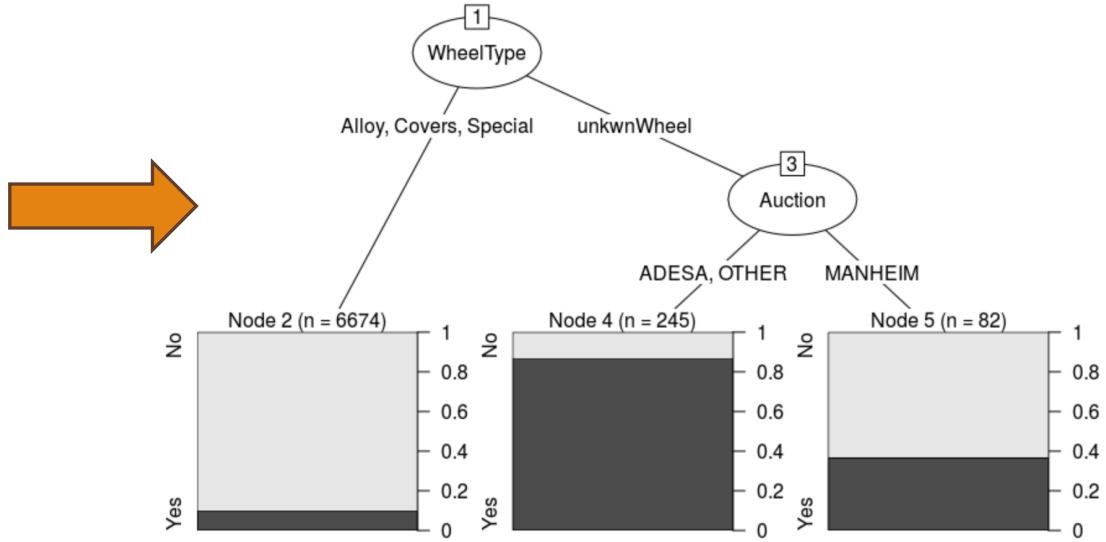
70% training data

30% testing data

Evaluate Decision Tree Model Performance: Training

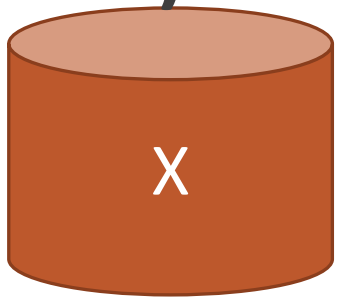
- Train decision tree on **training data (70%)**

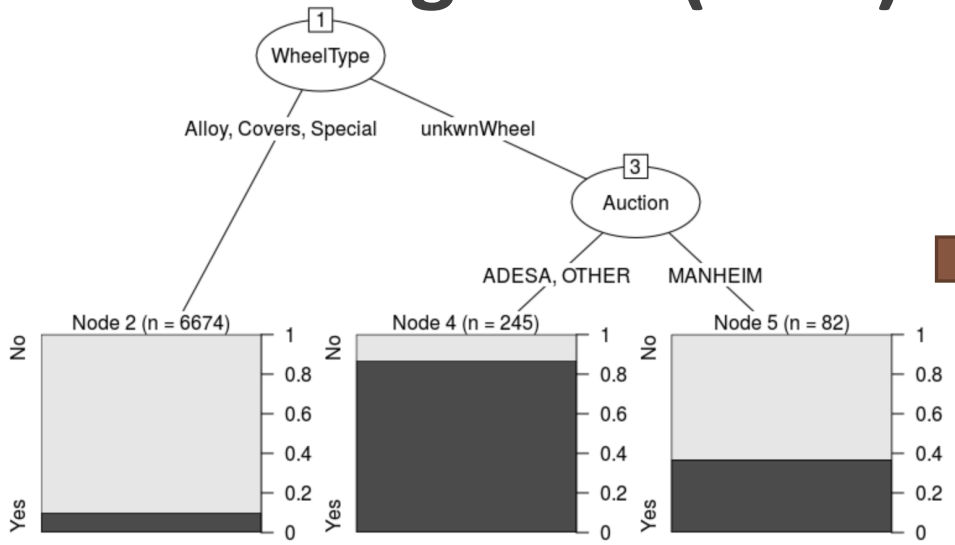
Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	WHITE	No	2871	LARGE TRUC	FORD	5300	8	75419	869	Alloy
ADESA	GOLD	Yes	1840	VAN	FORD	3600	8	82944	2322	Alloy
ADESA	RED	No	8931	SMALL SUV	CHRYSLER	7500	4	57338	588	Alloy
ADESA	GOLD	No	8320	CROSSOVER	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



Evaluate Decision Tree Model Performance: Testing

- Make predictions on **training data (70%) and 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: Generating Evaluation Metrics

- 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 (multiple choice + short answer)

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

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

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
- $P(A_i|C)$
  - If A_i is a categorical variable
    - $P(\text{Auction=OTHER}|\text{IsBadBuy=Yes}) = \frac{P(\text{Auction=OTHER, IsBadBuy=Yes})}{P(\text{C=Yes})}$
  - 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

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

- 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

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

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

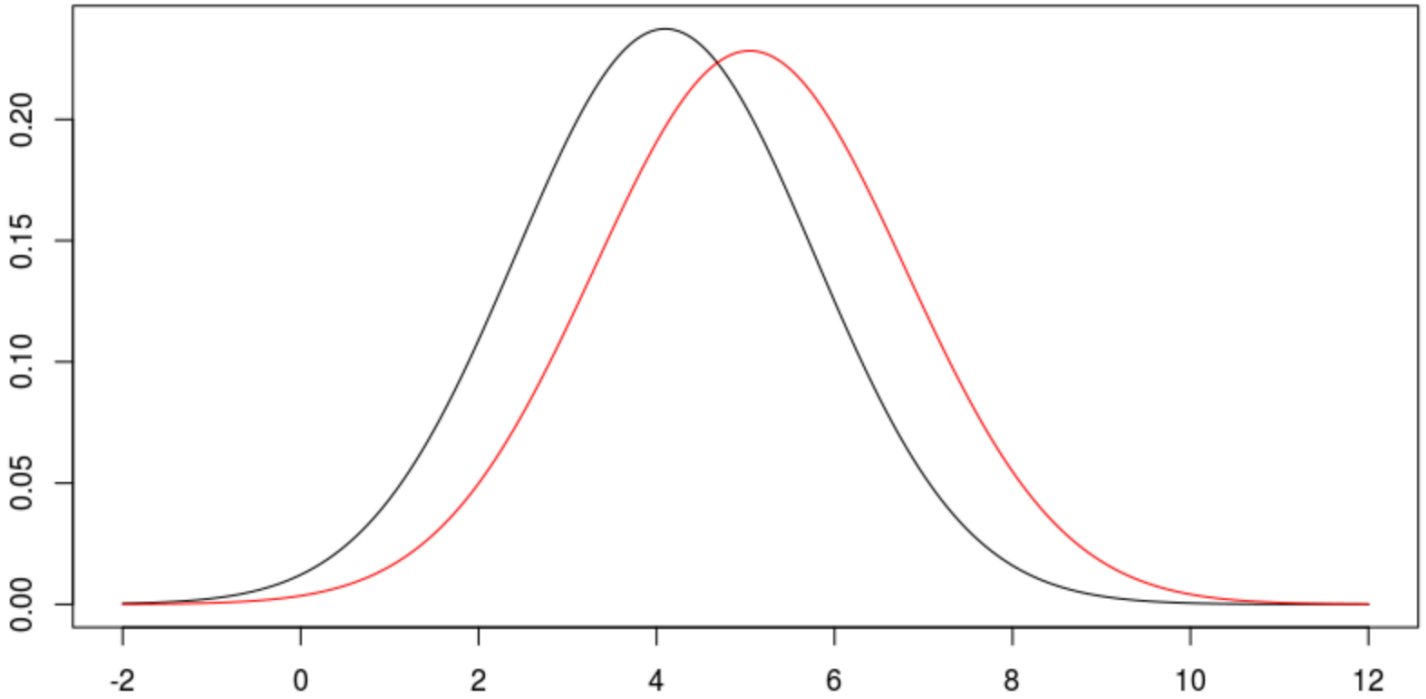
Naïve Bayes

- $\mathrm{P}(A_i|\mathrm{C})$: 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 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: Training

- 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

Prior

Conditional probabilities

```
A-priori probabilities:
Y
       No       Yes
0.8704471 0.1295529

Conditional probabilities:
     Auction
Y          ADESA    MANHEIM      OTHER
  No  0.1912416 0.5643759 0.2443825
  Yes 0.2340659 0.5439560 0.2219780

     MMRCurrentAuctionAveragePrice
Y          [,1]     [,2]
  No  6212.000 2374.427
  Yes 5232.111 2486.714

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

Naïve Bayes Evaluation: Testing

- 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: Generating Evaluation Metrics

- 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

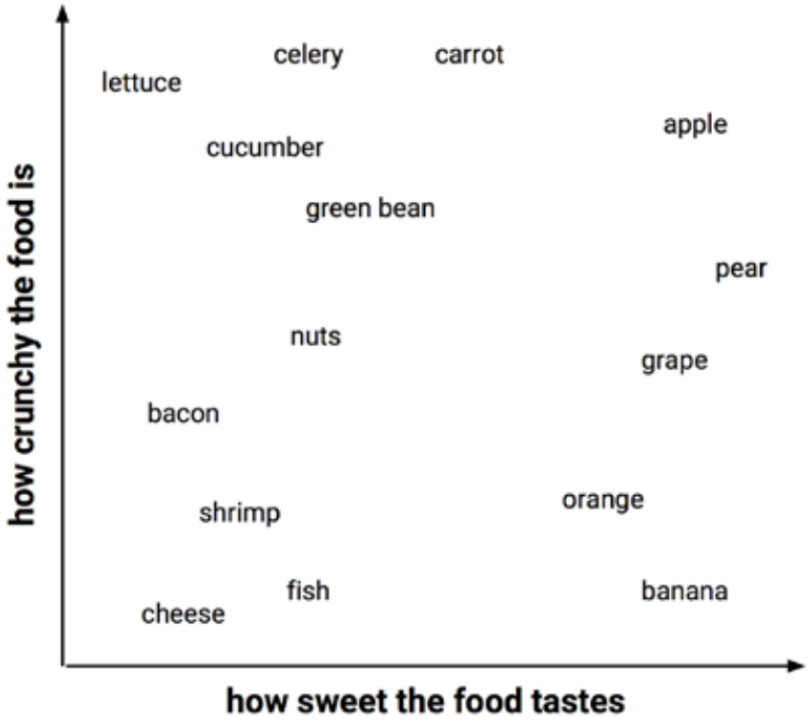
K Nearest Neighbor Algorithm (multiple choice + short answer)

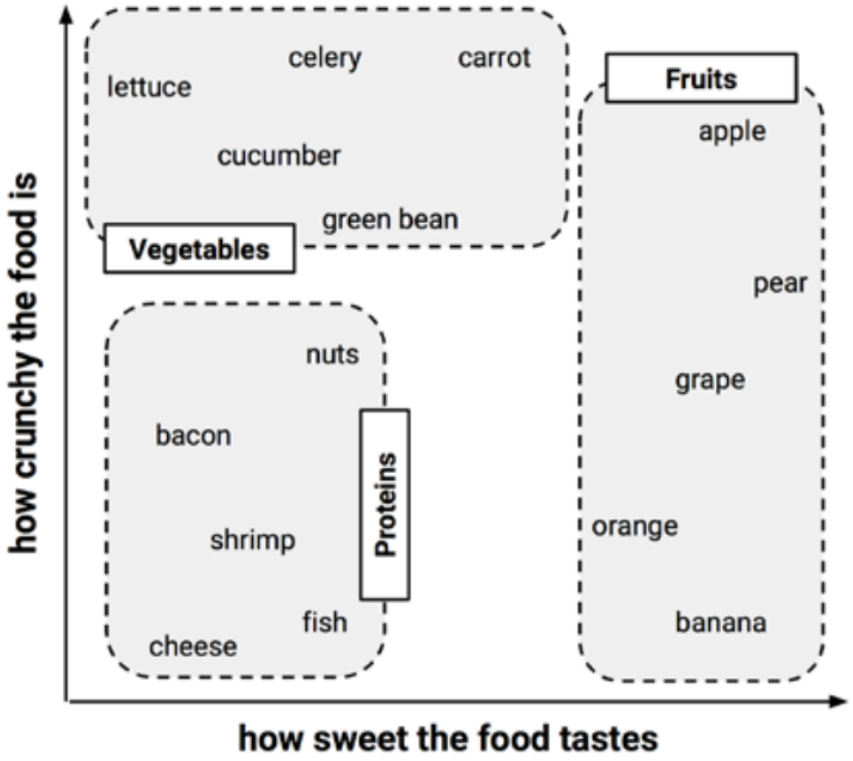
- Dark Dining
  - Suppose that prior to eating the mystery meal we had created a dataset in which we recorded our impressions of a number of ingredients we tasted previously.
  - To keep things simple, we rated only two features of each ingredient.
    - The first is a measure from 1 to 10 of how **crunchy** the ingredient is.
    - The second is a 1 to 10 score of how **sweet** the ingredient tastes.
  - We labeled each ingredient as one of the three types of food: fruit, vegetable, or protein.

Ingredient	Sweetness	Crunchiness	Food type
apple	10	9	fruit
bacon	1	4	protein
banana	10	1	fruit
carrot	7	10	vegetable
celery	3	10	vegetable
cheese	1	1	protein

K Nearest Neighbor Algorithm

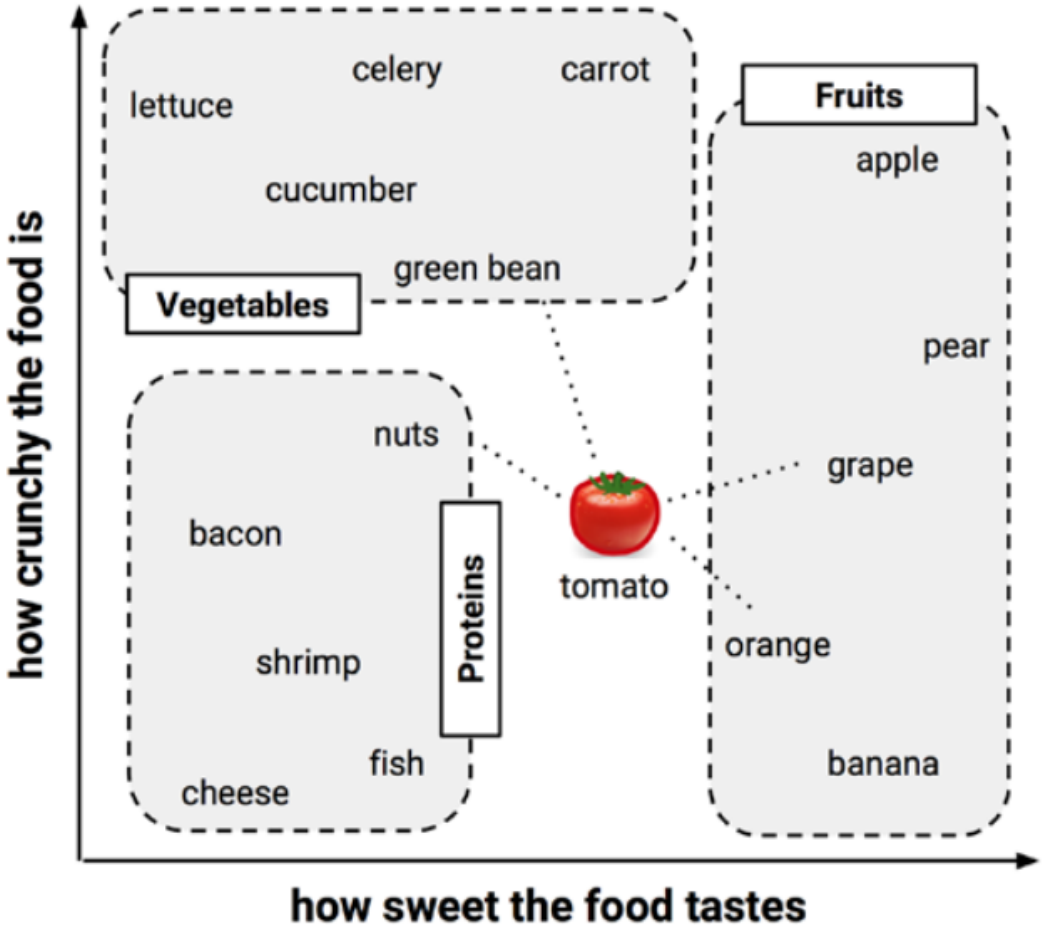
- Similar types of food tend to be grouped closely together





K Nearest Neighbor Algorithm

- Suppose you are given a tomato: protein, fruit or vegetable?
- We can use the nearest neighbor approach to determine which class is a better fit.

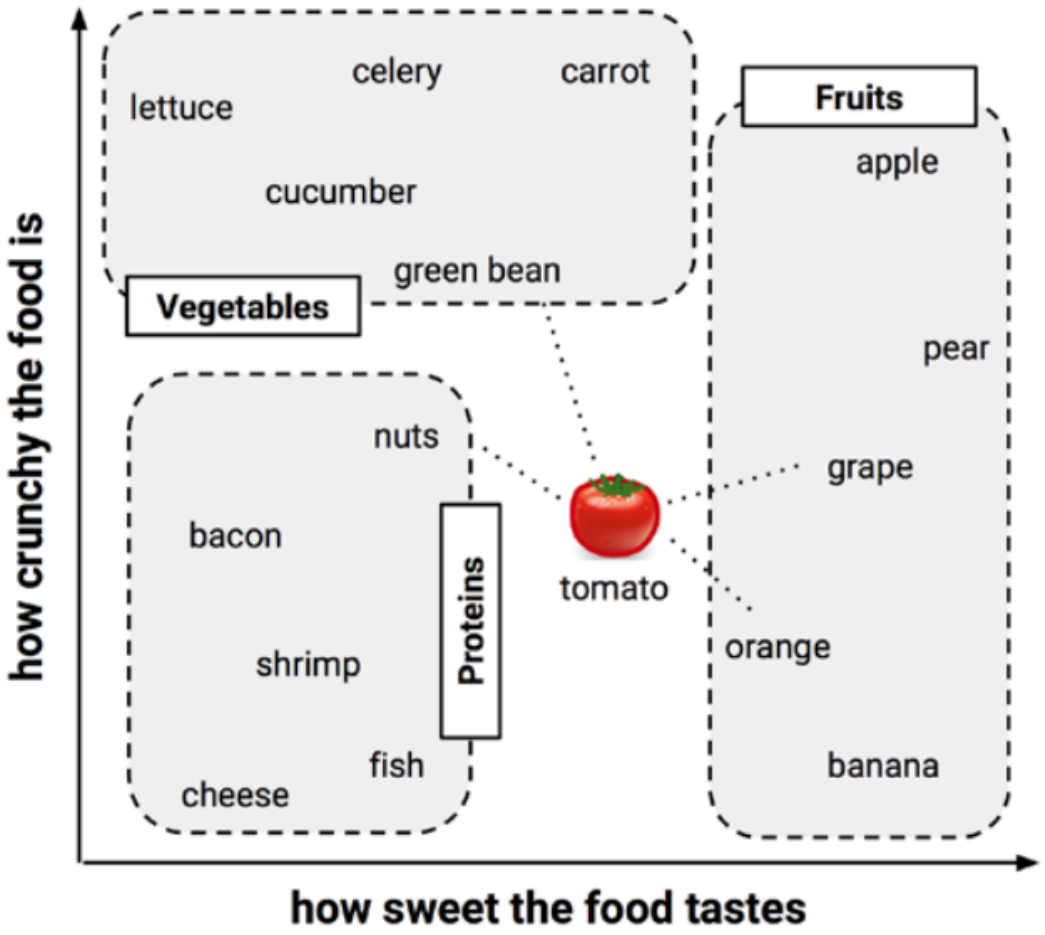


Measuring Similarity with Distance

- There are many different ways to calculate distance. The most commonly used measures are
  - Manhattan distance
  - Euclidean distance

Measuring Similarity with Distance

- Traditionally, the k-NN algorithm uses Euclidean distance.
  - For example, to calculate the distance between tomato *(sweetness = 6, crunchiness = 4)* and several of its closest neighbors:



Measuring Similarity with Distance

- Traditionally, the k-NN algorithm uses Euclidean distance.
  - For example, to calculate the distance between tomato *(sweetness = 6, crunchiness = 4)* and several of its closest neighbors:

Ingredient	Sweetness	Crunchiness	Food type	Distance to the tomato
grape	8	5	fruit	*sqrt((6 - 8)^2 + (4 - 5)^2) = 2.2*
green bean	3	7	vegetable	*sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2*
nuts	3	6	protein	*sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6*
orange	7	3	fruit	*sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4*

Choosing an Appropriate k

- To classify the tomato as a vegetable, protein, or fruit, we'll begin by assigning the tomato, the food type of its single nearest neighbor.
  - This is called 1-NN classification because k = 1.
  - The orange is the nearest neighbor to the tomato, with a distance of 1.4. As orange is a fruit, the 1-NN algorithm would classify tomato as a fruit.
- If we use the k-NN algorithm with k = 3 instead, it performs a vote among the three nearest neighbors: orange, grape, and nuts.
  - Since the majority class among these neighbors is fruit (two of the three votes), the tomato again is classified as a fruit.

Ingredient	Sweetness	Crunchiness	Food type	Distance to the tomato
grape	8	5	fruit	*sqrt((6 - 8)^2 + (4 - 5)^2) = 2.2*
green bean	3	7	vegetable	*sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2*
nuts	3	6	protein	*sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6*
orange	7	3	fruit	*sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4*

Choosing an Appropriate k

- The decision of how many neighbors to use for k-NN determines how well the model will generalize to future data.
  - Choosing a large *k* reduces the impact or variance caused by noisy data, but can bias the learner so that it runs the risk of ignoring small, but important patterns. UNDERFIT K = total number of instances in the training data
  - On the opposite extreme, using a single nearest neighbor allows the noisy data or outliers to unduly influence the classification of examples. OVERFIT K = 1

Most common category

Preparing Data for Use with K-NN

- Features are typically transformed to a **standard range** prior to applying the k-NN algorithm.
  - Each feature contributes relatively equally to the distance formula
  - **min-max normalization**

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

  - **z-score standardization**

$$X_{new} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{StdDev}(X)}$$

Preparing Data for Use with K-NN

NAME	AGE	SPENDING($)
SUE	21	2300
CARL	27	2600
TOM	32	5400
JACK	35	6000
DAN	44	6200
JILL	50	7000

NAME	AGE	SPENDING($)
SUE	0	0
CARL	0.207	0.064
TOM	0.379	0.660
JACK	0.483	0.787
DAN	0.793	0.830
JILL	1	1

- Euclidean distance (Tom, Jack)=

$$\sqrt{(0.379-0.483)^2+(0.660-0.787)^2}=0.164$$

Preparing Data for Use with K-NN

- The Euclidean distance is not defined for categorical variables.
  - Convert categorical variables into numeric format – **dummy coding**
    - **Create dummy variables to replace the original categorical variable.**
    - Dummy coding for a binary variable: gender

$$\text{male} = \begin{cases} 1 & \text{if x} = \text{male} \\ 0 & \text{otherwise} \end{cases}$$

    - Dummy coding for n-category variable: temperature variable (for example, hot, medium, or cold)

$$\text{hot} = \begin{cases} 1 & \text{if x} = \text{hot} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{medium} = \begin{cases} 1 & \text{if x} = \text{medium} \\ 0 & \text{otherwise} \end{cases}$$

We need n-1 dummy variables

Preparing Data for Use with K-NN

NAME	AGE	SPENDING($)	GENDER
SUE	21	2300	F
CARL	27	2600	M
TOM	32	5400	M
JACK	35	6000	M
DAN	44	6200	M
JILL	50	7000	F

NAME	AGE	SPENDING($)	GENDER
SUE	0	0	1
CARL	0.207	0.064	0
TOM	0.379	0.660	0
JACK	0.483	0.787	0
DAN	0.793	0.830	0
JILL	1	1	1

K Nearest Neighbor 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

K Nearest Neighbor Evaluation: Training

- Train K Nearest Neighbor 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

K-NN Stores the training data

K Nearest Neighbor Evaluation: Testing

- Make predictions on **testing data (30%)** and **training data (70%)**
  - Search through its similarities with training instances to find *k* nearest neighbors in the training set.
  - Estimate its target variable value based on *k* nearest neighbors' known target values.
    - Binary Classification – e.g. majority class of *k* nearest neighbors

K Nearest Neighbor Evaluation: Testing

Auction	Color	IsBadBuy	MMRCurrentAu	Size	TopThreeAm	VehBCost	VehicleAge	VehOdo	WarrantyCos	WheelType
ADESA	RED		7860	MEDIUM	CHRYSLER	7500	2	50644	754	Covers

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 similarities

find *k* nearest neighbors in the training set

Make prediction: majority class of *k* nearest neighbors

K Nearest Neighbor

- Training phase:
  - Import and store training data in the “memory”
  - No actual “model” is built
  - Fast but memory intensive
- Prediction phase:
  - The main work and time consuming
- Common characterization
  - Instanced-based learning algorithms
  - Memory-based learning algorithms
  - Lazy learners

Evaluation Method (multiple choice)

- 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!

Evaluation Method

- 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)

Splitting Method

- 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

Cross Validation

- Cross validation steps
  - Split the entire data randomly into *k* folds.
  - Then fit the model using the *k*-1 folds and validate/test the model using the *k*th fold. Store the evaluation results.
  - Repeat this process until every fold serve as the test set. Then take the average of your recorded evaluation results.

Cross Validation

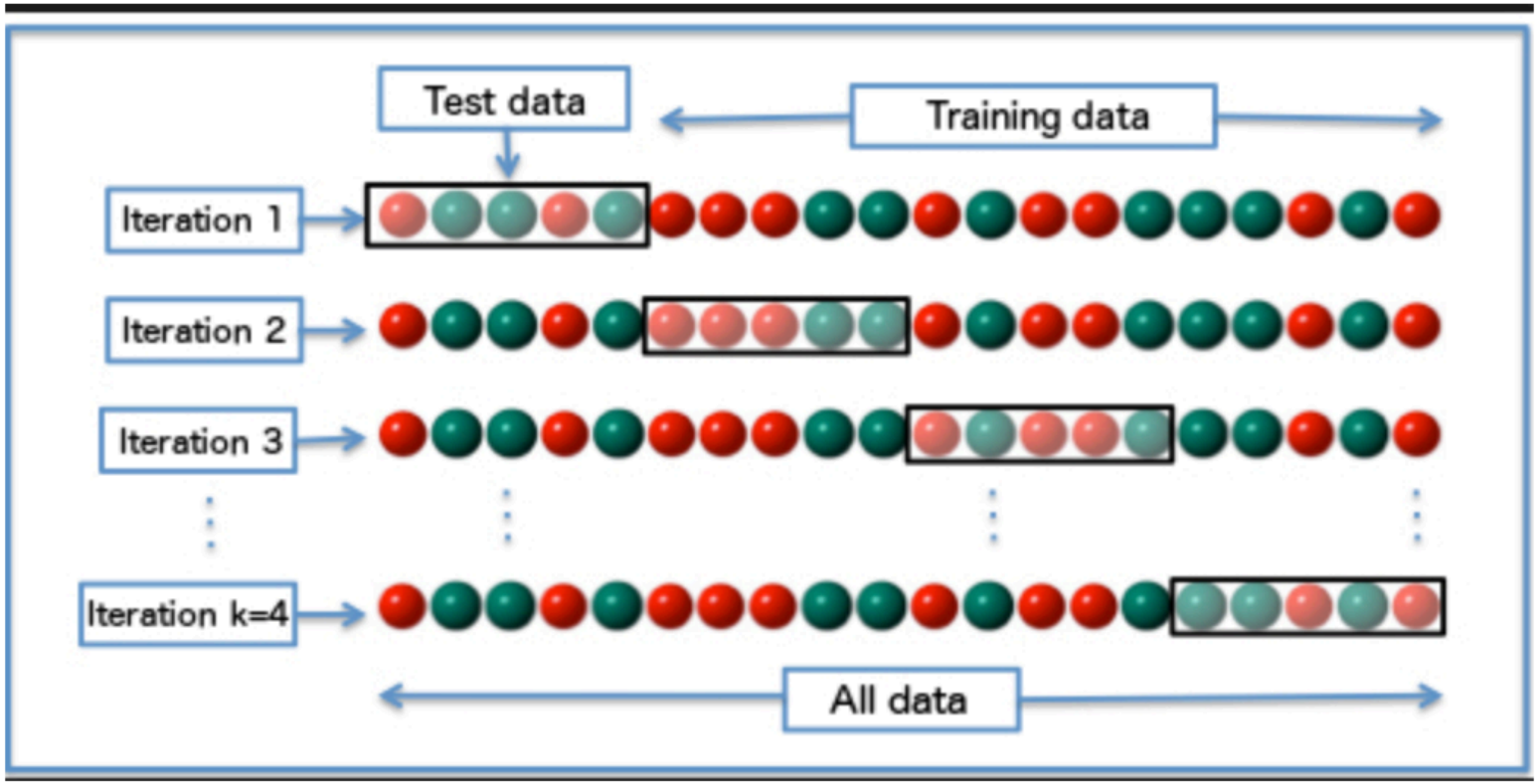


Image Sourced From Wikipedia

Training and testing will be done for ? times

Each instance will be used for training ? times

Each instance will be used for testing ? times

Evaluation Metrics Method (multiple choice + short answer)

- 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**.

Evaluation Metrics

	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

Evaluation Metrics

- **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 Metrics

- Evaluate performances on positive/negative 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

	pred	
target	No	Yes
No	2601	10
Yes	302	86

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

Causes for Overfitting:

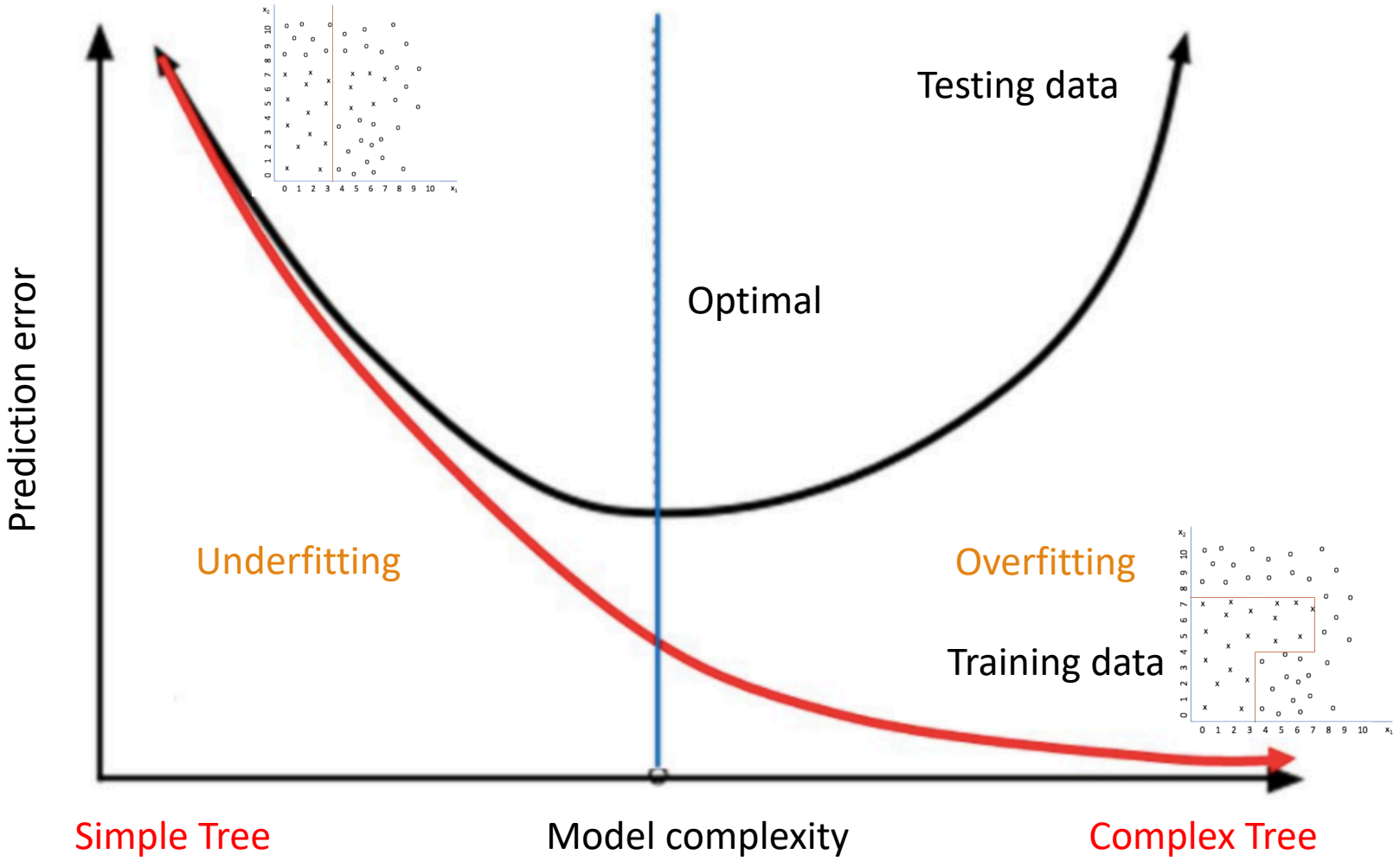
- Training data is not a good representation of testing (new) data
  - Insufficient training data
  - Noises in data: inconsistent class labels for the same values in feature set (input attributes)
  - Outliers in data: the number of samples with a given combination of class labels and feature values 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



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

- 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

  - 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	2601	10
Yes	302	86

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

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

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	2601	10
Yes	302	86

Naïve Bayes Model

	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

  - 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	pred Yes
No	2601	10
Yes	302	86

target	pred No	pred Yes
No	2490	121
Yes	287	101

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

Evaluation decisions

- How to extract training and holdout testing data sets from one collection of observations?
- When is model performance acceptable?
- How to improve underfitted or overfitted models?
- Which evaluation metric or metrics matter?
- Why is overall accuracy not the best measure?
- Why would we be interested in performance measures of different classes?
- Would a class be more important than other classes?
- When is precision/recall important?
- Why are we interested in F-measure of a class?

Exam Questions

- 21 questions
  - 15 multiple choice/True false/multiple answers
  - 5 short answers
    - Decision tree (labs and quiz)
    - Naïve Bayes (labs and quiz)
    - kNN (differences between kNN and other methods)
    - Evaluation results (generalization, precision, recall, F-measure, performance comparison: majority/minority class, different models)
    - Business problem
  - 1 code writing