Customer Analytics
Review & Practice session

Professor Vincent Nijs

Rady School of Management @ UCSD

Customer Analytics

Reminders

- Peer evaluations for the PFG-bank presentations
- Intra-group peer evaluations: https://rsm-compute-01.ucsd.edu:4443/peer_eval/
- Exam will be in-person and run from 2pm – 5pm PT on Thursday 3/24
- Come to room 4N128 to find out if you will be in that room of 3N128

Final exam: What you can expect

- Causality check-lists
- Customer Lifetime Value calculations (CLV)
- Manipulate data (e.g., transform variables, ‘bin’ a continuous variable)
- Exploratory Data Analysis (EDA)
- Linear and Logistic regression (incl prediction plots)
- Evaluate relative importance of explanatory variables (features) from an AD or ML model (permutation importance plots)
- Estimate interactions and generate plots from ML models to identify if an interaction exists
- Use training and test samples
- Generate predictions using Linear/Logistic regression, NN, Random Forests, XGBoost, etc.
- Create lift, gains, and profit charts and evaluate overfitting
- Determine profits and return on marketing expenditures
- Uplift modeling
- Estimate logistic regression on experimental data
- Bias-Variance tradeoff
- Tuning ML models using Cross-Validation
- Understand benefits and limitations of partial factorial design
- ...

Goals for Review session

Complete during session:

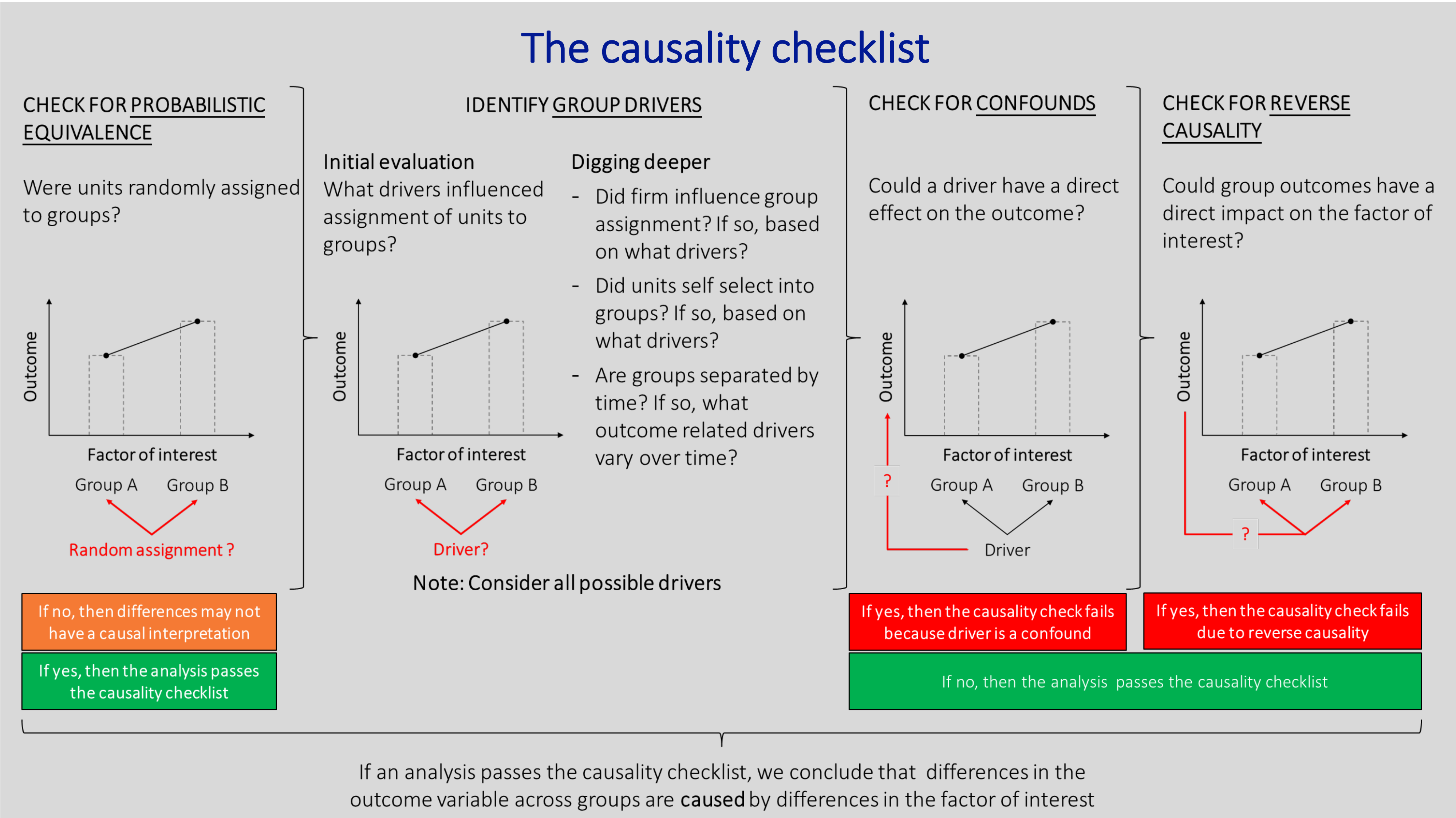
- Task 1: Causality check list (2 in-class, 2 pre-work)
- Task 2: Linear regression
- Task 3: Logistic regression
- Task 4: (facebook.ipynb | interactions)
- Task 8: (bizware-review.ipynb | experimental design and logistic regression)
- Task 9: (impurity calculations)

Discuss during session:

- Task 7: (slow-auc.ipynb)
- Task 8: (bbb_sklearn.ipynb)

Extra:

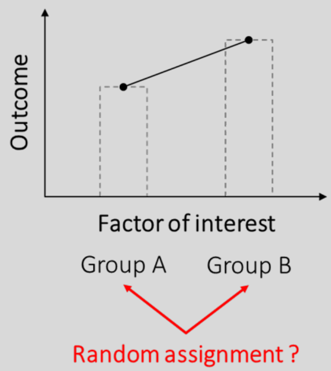
- Task 5: Customer Lifetime Value calculation (clv.xlsx and clv.ipynb)


The causality checklist
CHECK FOR PROBABILISTIC EQUIVALENCE
Were units randomly assigned to groups?
Outcome
Factor of interest
Group A
Group B
Random assignment ?
If no, then differences may not have a causal interpretation
If yes, then the analysis passes the causality checklist
IDENTIFY GROUP DRIVERS
Initial evaluation
What drivers influenced assignment of units to groups?
Outcome
Factor of interest
Group A
Group B
Driver?
Digging deeper
- Did firm influence group assignment? If so, based on what drivers?
- Did units self select into groups? If so, based on what drivers?
- Are groups separated by time? If so, what outcome related drivers vary over time?
Note: Consider all possible drivers
CHECK FOR CONFOUNDS
Could a driver have a direct effect on the outcome?
Outcome
Factor of interest
Group A
Group B
?
Driver
If yes, then the causality check fails because driver is a confound
CHECK FOR REVERSE CAUSALITY
Could group outcomes have a direct impact on the factor of interest?
Outcome
Factor of interest
Group A
Group B
?
If yes, then the causality check fails due to reverse causality
If no, then the analysis passes the causality checklist
If an analysis passes the causality checklist, we conclude that differences in the outcome variable across groups are caused by differences in the factor of interest

Review the Google Ads example

CHECK FOR PROBABILISTIC EQUIVALENCE

Were units randomly assigned to groups?

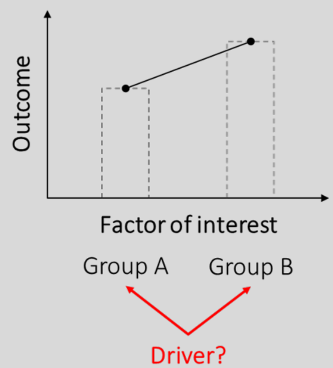


If no, then differences may not have a causal interpretation

If yes, then the analysis passes the causality checklist

IDENTIFY GROUP DRIVERS

What drivers influenced assignment of units to groups?

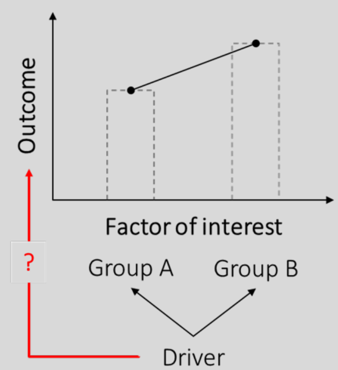


List drivers …

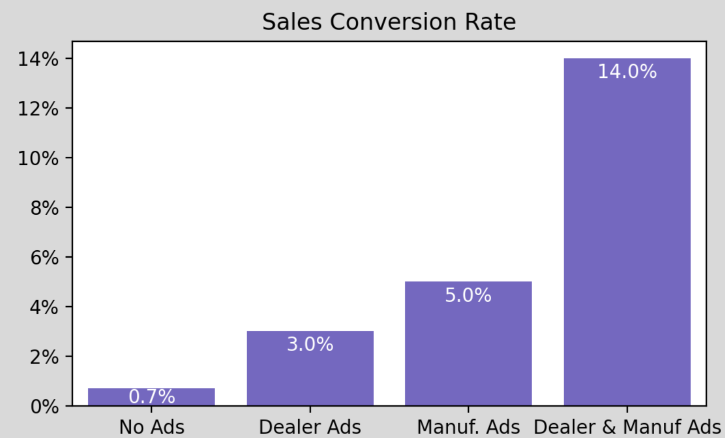
- Google search terms
- Interest in buying a car

CHECK FOR CONFOUNDS

Could a driver have a direct effect on the outcome?



The driver is a confound



Summary of insights from applying the causality checklist to the Google ads example

- **Causal claims**: (1) Google ads work, (2) Retailer and Manufacturer ads are complements
- **Factor of interest**: Car ad exposure
- **Groups**: Customers that saw (1) no ads, (2) retailer ads, (3) manufacturer ads, and (3) retailer and manufacturer ads
- **Outcome**: Sales conversion
- **Group assignment**: Not random
- **Drivers**: Google search terms reflecting interest in buying a car
- **Confound**: Yes. Interest in buying a car can have a direct effect on the likelihood of buying a car

TASK 1: Apply the checklist to evaluate causal statements

A company sells many snowmobiles in Canada but very few in Mexico. The company also advertises extensively in Canada but does not advertise at all in Mexico

Causal claim: Advertising works

Factor of interest:

Groups:

Outcome:

Group assignment:

Drivers of group assignment:

Confound:

Reverse causality:

TASK 1: Apply the checklist to evaluate causal statements

Snowmobile sales are below expectations in January and a dealership in Toronto plans to run a promotion in February. During the promotional period an unexpected snow-storm hits the Toronto area. Sales of snowmobiles in February are 10% higher than expected

Causal claim: The promotion caused a 10% increase in sales

Factor of interest:

Groups:

Outcome:

Group assignment:

Drivers of group assignment:

Confound:

Reverse causality:

TASK 1: Apply the checklist to evaluate causal statements

Doordash is a logistics software startup. Affiliated drivers deliver restaurant food to customers. A restaurant decides to put a link to Door Dash on their website, starting in January. The number of orders for take-out in January are 5% lower than in December

Causal claim: Adding the link to the Door Dash site caused a decrease in sales

Factor of interest:

Groups:

Outcome:

Group assignment:

Drivers of group assignment:

Confound:

Reverse causality?:

TASK 1: Apply the checklist to evaluate causal statements

A manufacturer of kitchen knives has improved the quality of their product each year. The company also increased prices each year to cover the costs of these quality improvements. A regression of price on demand (i.e., demand = a + b x price) gives a coefficient for price very close to 0 that is not statistically significant

Causal claim: Customers are not sensitive to price changes so the manufacturer can continue to increase prices, even if quality is not improved

Factor of interest:

Groups:

Outcome:

Group assignment:

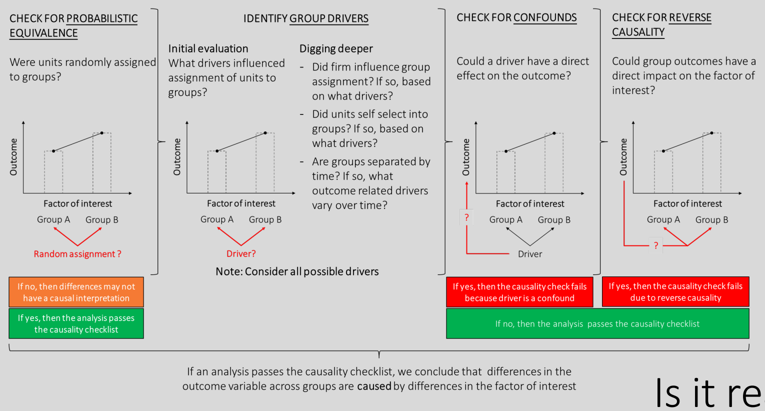
Drivers of group assignment:

Confound:

Reverse causality?:

Getting from prediction to prescription requires careful deliberation

PREDICTION → PRESCRIPTION CHECKLIST



Is the data available for prediction relevant for the desired prescription?

Yes

No

Gather additional data or experiment

Is it reasonable to attach a causal interpretation to the estimated effect of the prescription?

- Based on Exhibit 1?
- Based on Testing?

Yes

No

✔

Experiment or fix the analysis

Task 2: Regression review (see linear-regression.ipynb)

```
1  rsm.coef_ci(reg)
```

✓ 0.0s Python

	index	coefficient	2.5%	97.5%	p.values	
0	Intercept	-6780.993	-7182.855	-6379.131	< .001	***
1	clarity[T.SI2]	2790.760	2395.873	3185.646	< .001	***
2	clarity[T.SI1]	3608.531	3215.384	4001.679	< .001	***
3	clarity[T.VS2]	4249.906	3854.604	4645.208	< .001	***
4	clarity[T.VS1]	4461.956	4060.801	4863.111	< .001	***
5	clarity[T.VVS2]	5109.476	4697.311	5521.640	< .001	***
6	clarity[T.VVS1]	5027.669	4607.574	5447.764	< .001	***
7	clarity[T.IF]	5265.170	4807.024	5723.317	< .001	***
8	carat	8438.030	8337.834	8538.227	< .001	***

Click Ball Point Pens

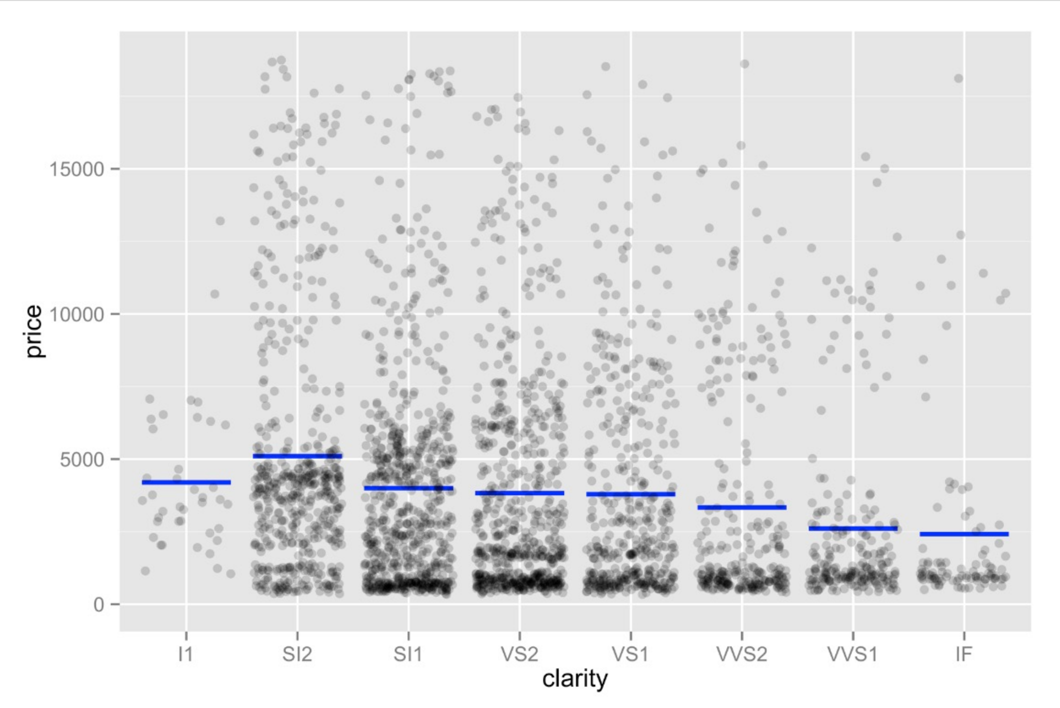
- **Company:** A national manufacturer of ball point pens.

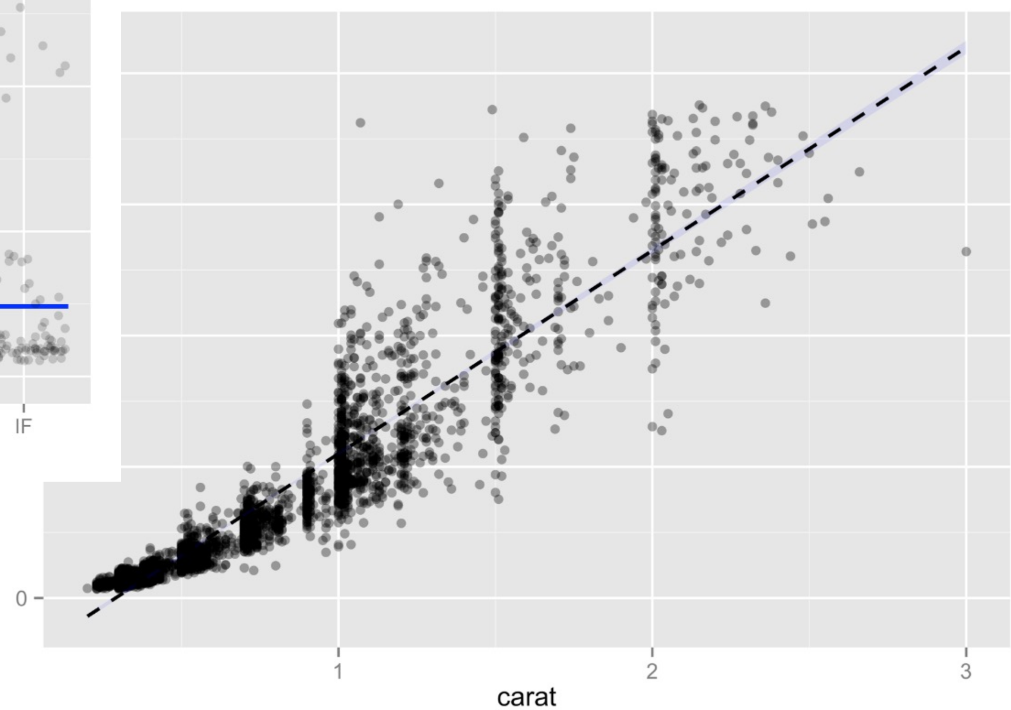
- **Managerial problem:**
  - What is the value of an advertising spot?
  - How much should we pay sales reps?
  - Are the results the same when you include both advertising and sales reps in the model? If not, why not?

- **Data:** Sales data for 40 markets/territories along with measures of marketing effort

- Use **linear-regression.ipynb** and **data/click.pkl**

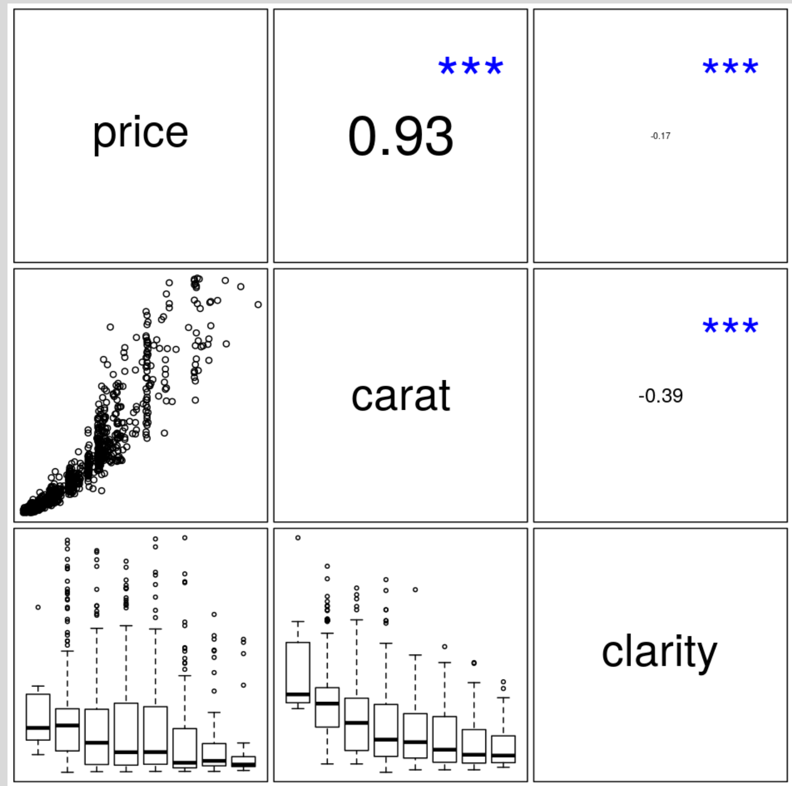
Omitted Variable Bias (OVB)





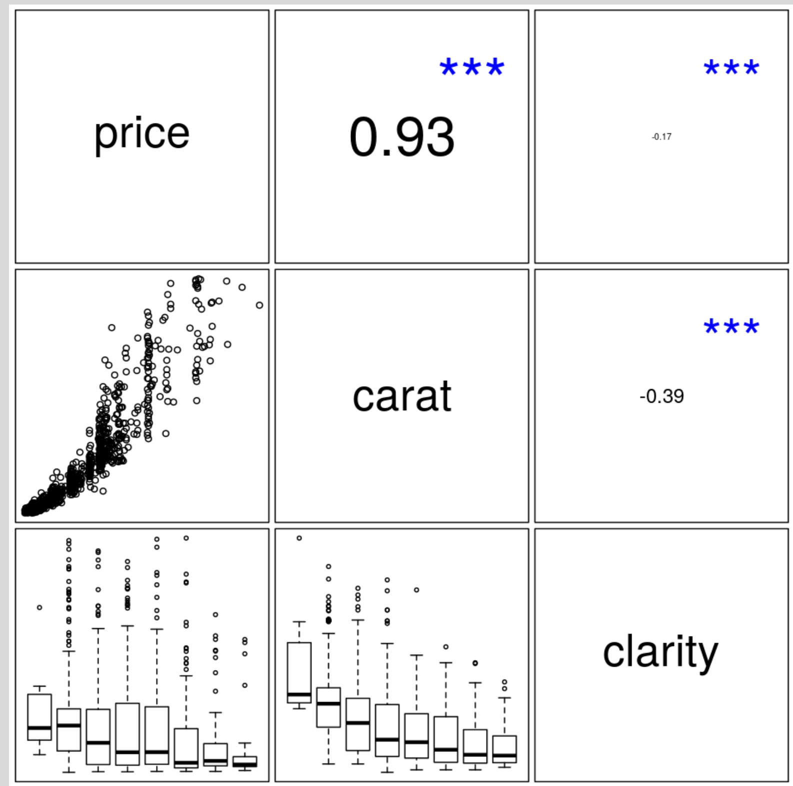
Omitted Variable Bias (OVB) and Multi-collinearity (MC)

```
              coefficient std.error t.value p.value
 (Intercept)     4194.775   616.530   6.804  < .001 ***
 clarity|SI2      905.414   639.415   1.416   0.157
 clarity|SI1     -196.198   633.401  -0.310   0.757
 clarity|VS2     -371.808   634.911  -0.586   0.558
 clarity|VS1     -405.594   643.823  -0.630   0.529
 clarity|VVS2    -856.955   658.518  -1.301   0.193
 clarity|VVS1   -1586.315   669.318  -2.370   0.018 *
 clarity|IF     -1783.078   730.540  -2.441   0.015 *

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1

R-squared: 0.031,  Adjusted R-squared: 0.029
F-statistic: 13.759 df(7,2992), p.value < .001
Nr obs: 3,000
```



Omitted Variable Bias (OVB) and Multi-collinearity (MC)

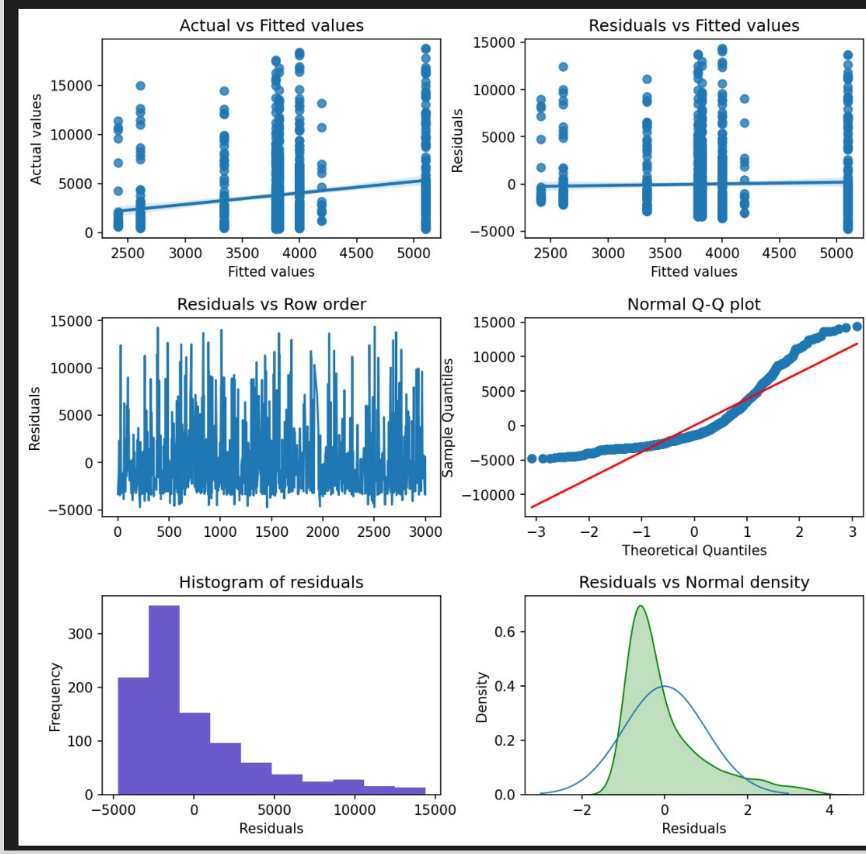
```
               coefficient std.error t.value p.value
 (Intercept)     -6780.993   204.952 -33.086  < .001 ***
 carat            8438.030    51.101 165.125  < .001 ***
 clarity|SI2      2790.760   201.395  13.857  < .001 ***
 clarity|SI1      3608.531   200.508  17.997  < .001 ***
 clarity|VS2      4249.906   201.607  21.080  < .001 ***
 clarity|VS1      4461.956   204.592  21.809  < .001 ***
 clarity|VVS2     5109.476   210.207  24.307  < .001 ***
 clarity|VVS1     5027.669   214.251  23.466  < .001 ***
 clarity|IF       5265.170   233.658  22.534  < .001 ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1

R-squared: 0.904,  Adjusted R-squared: 0.904
F-statistic: 3530.024 df(8,2991), p.value < .001
Nr obs: 3,000
```



Check residuals

```
1  rsm.reg_dashboard(reg)
```

✓ 0.6s Python



Model fit

```python
reg = smf.ols(formula="price ~ carat + clarity", data=diamonds).fit()
rsm.evalreg(diamonds.assign(pred_reg = reg.fittedvalues), "price", "pred_reg")
```

✓ 0.0s Python

	Type	predictor	n	r2	mse	mae
0	All	pred_reg	3000	0.904	1498982.488	833.179

```python
rsm.model_fit(reg)
```

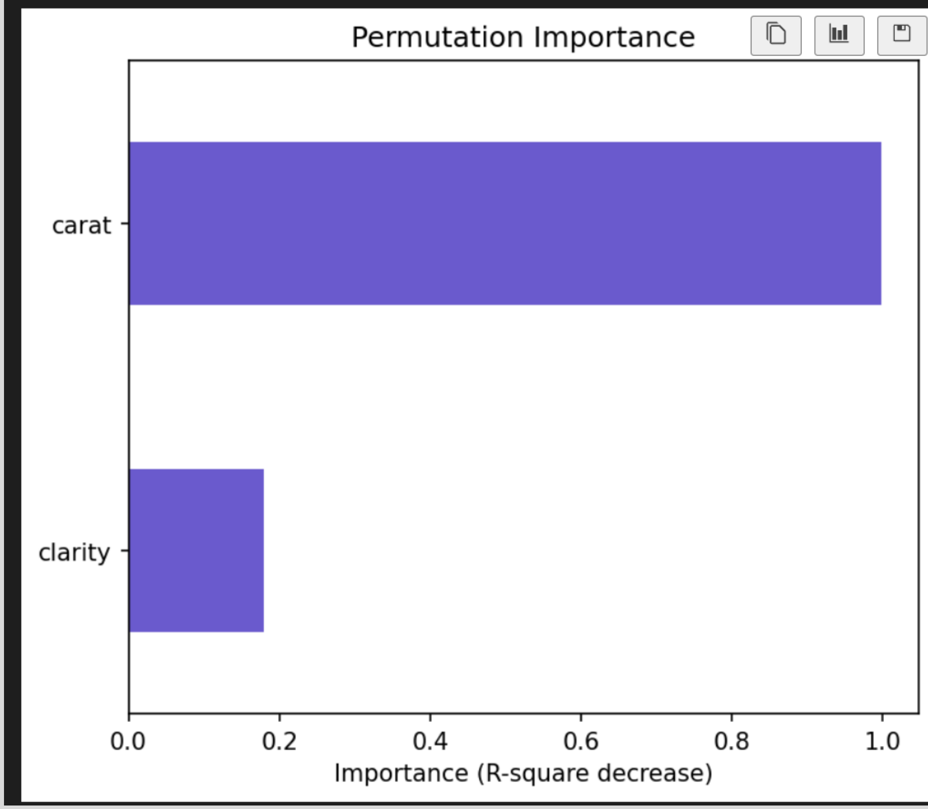
✓ 0.0s Python

```
R-squared: 0.904, Adjusted R-squared: 0.904
F-statistic: 3530.024 df(8, 2991), p.value < 0.001
Nr obs: 3,000
```

Variable importance

```python
reg = smf.ols(formula="price ~ clarity + carat", data=diamonds).fit()
rsm.vimp_plot_sm(reg, diamonds)
```

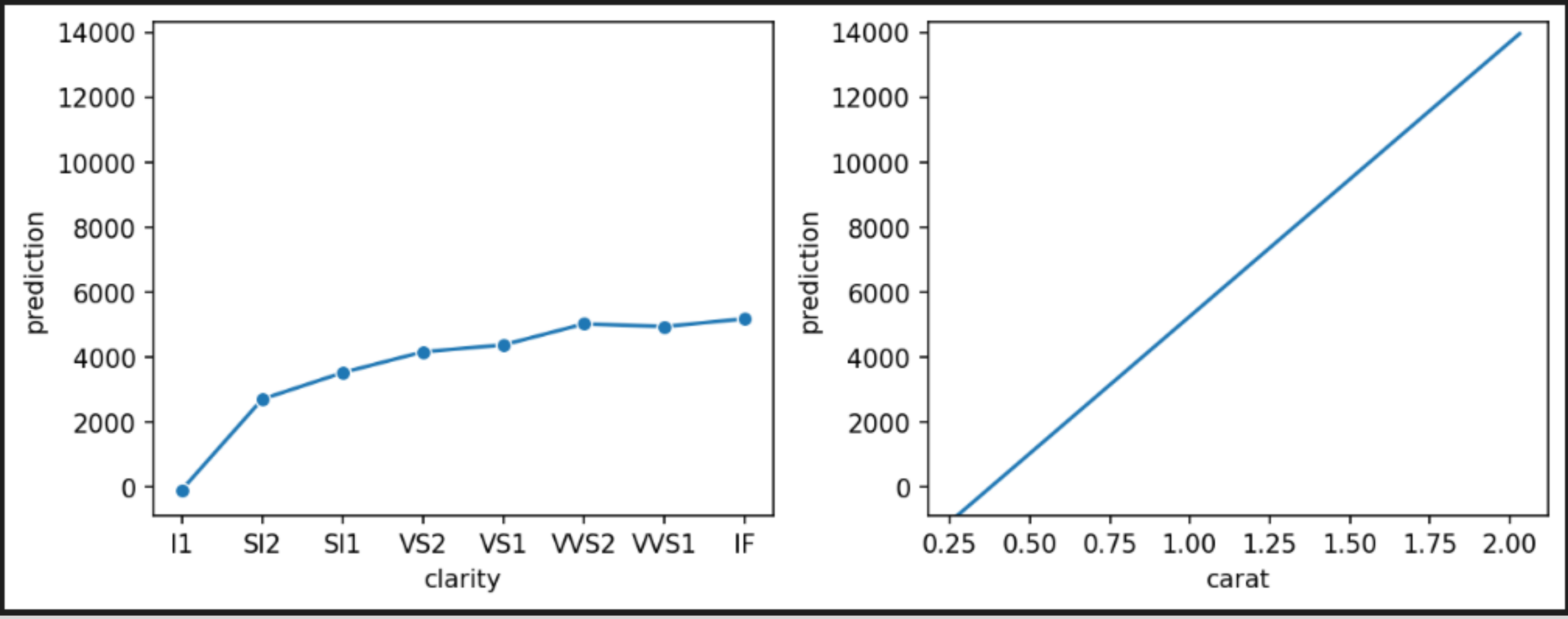
✓ 0.2s Python



Variable effect

```python
1  rsm.pred_plot_sm(reg, diamonds)
```

✓ 0.1s   Python



Task 3: Logistic regression (see logistic-regression.ipynb)

```
1  rsm.or_ci(lr)
```

✓ 0.0s Python

	index	OR	OR%	2.5%	97.5%	p.values	
1	coupon	2.169	116.9%	2.105	2.234	< .001	***
2	purch	1.095	9.5%	1.085	1.106	< .001	***
3	last	0.933	-6.7%	0.930	0.937	< .001	***

Model fit

```python
rsm.model_fit(lr)
```

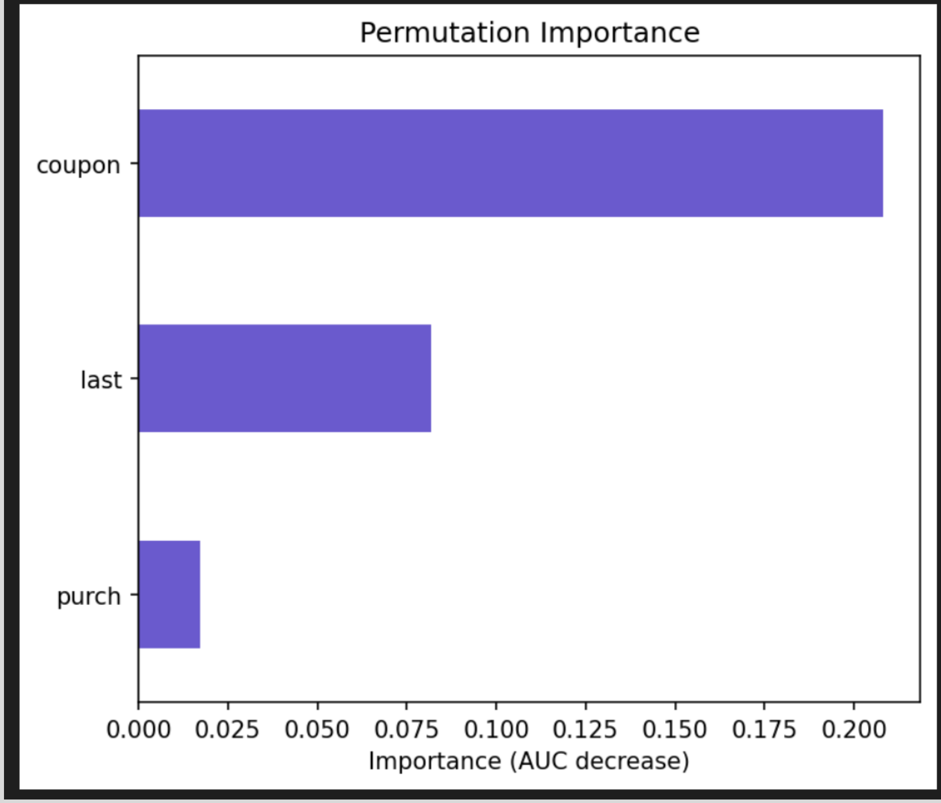
Python

```
Pseudo R-squared (McFadden): 0.208
Pseudo R-squared (McFadden adjusted): 0.208
Area under the RO Curve (AUC): 0.803
Log-likelihood: -9110.529, AIC: 18229.058, BIC: 18260.672
Chi-squared: 20007.878 df(3), p.value < 0.001
Nr obs: 20,000
```

Variable importance

```
rsm.vimp_plot_sm(lr, dvd)
```

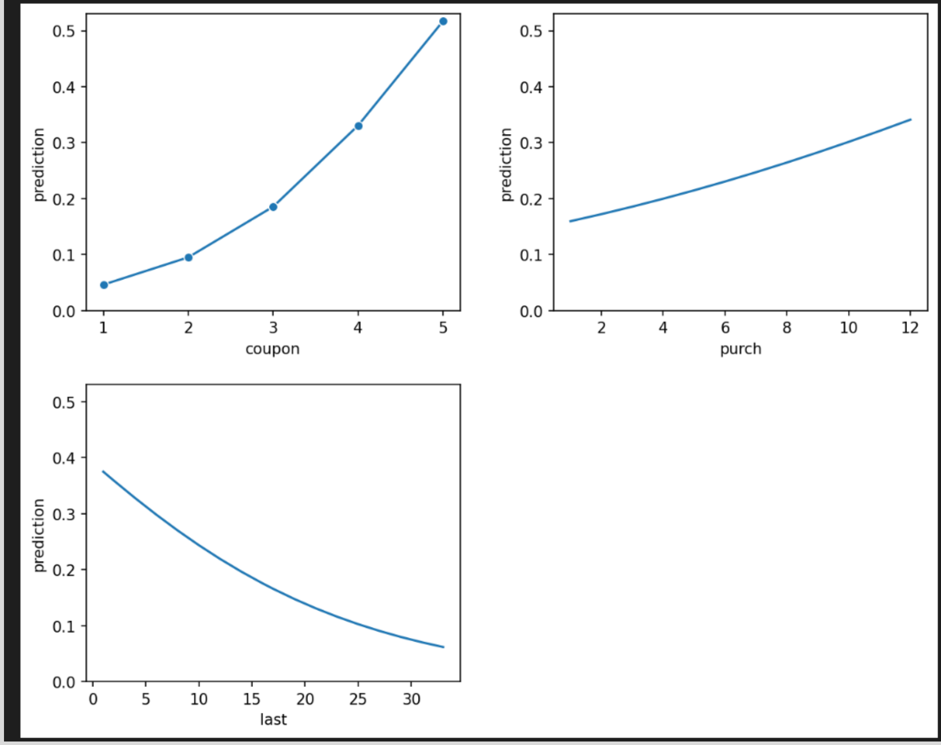
Python



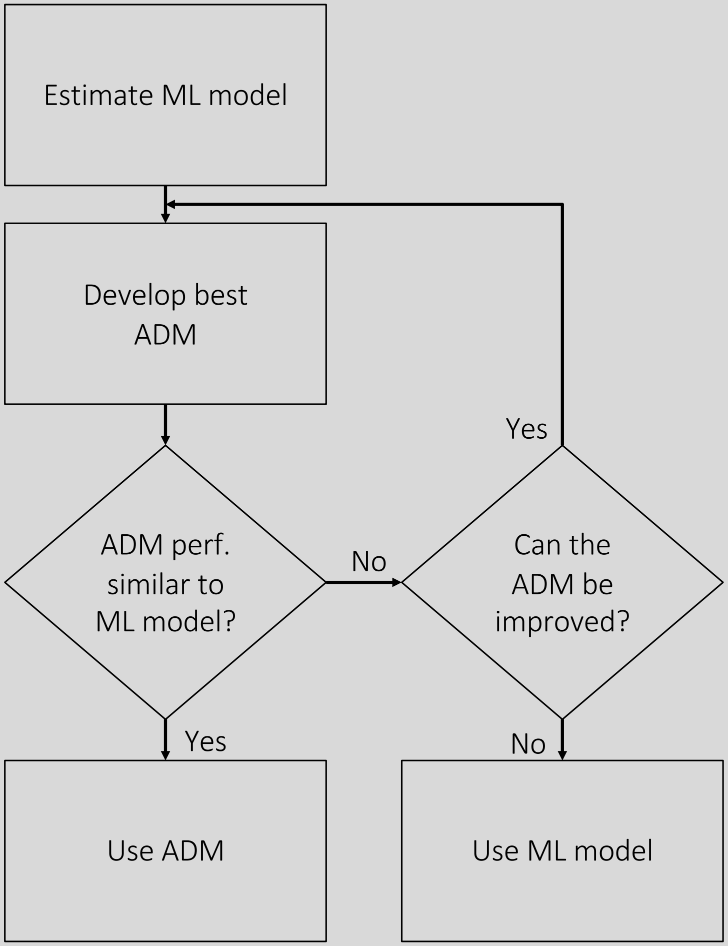
Variable importance

```python
rsm.pred_plot_sm(lr, dvd)
```

Python



Machine Learning (ML) models can be used in combination with Analyst Driven Models (ADM)

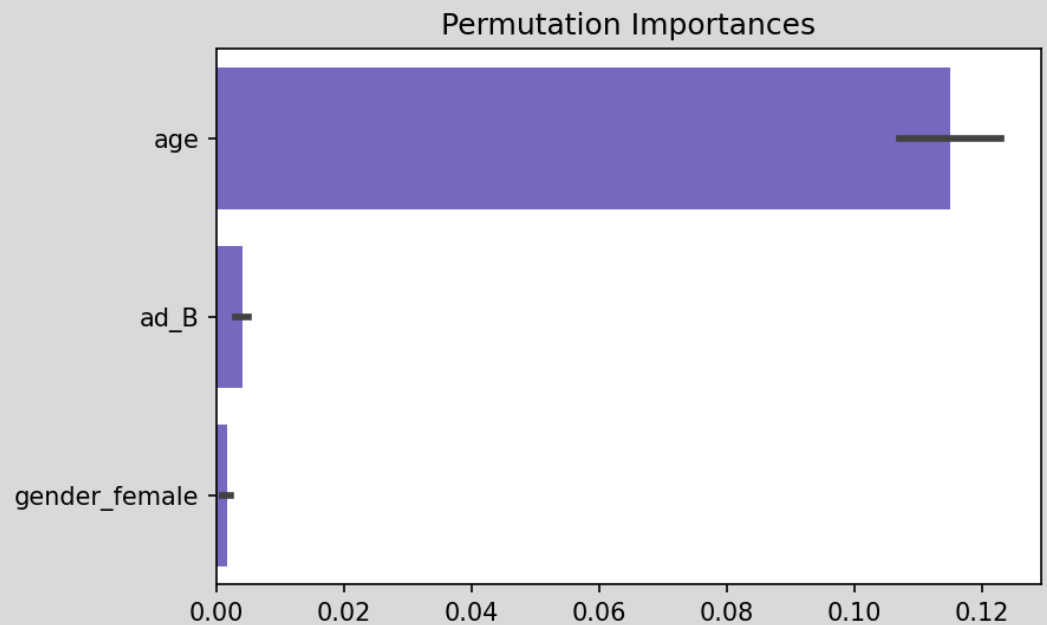


Core idea:

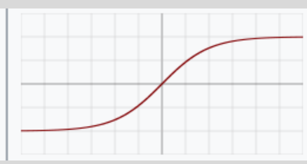
- Use ML model as performance benchmark
- Use ADM for interpretation

TASK 4: What predicts ad click-through? (see facebook.ipynb)

- Reproduce the plot on the right using a NN (1)
- How does the plot change as we add another node to the hidden layer, i.e., NN(2)? Why does it change?
- Develop a logistic regression model that achieves similar performance to the NN(2) model (use gainsplot)
- Use prediction plots to demonstrate key new effects are capture by the LR model



TanH



$$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

TASK 5: Calculate CLV (use task-5-clv.xlsx or task-5-clv.ipynb)

		Years				
	Start of CLV Calc.	1	2	3	4	5
Revenues	$0	$400	$400			
Product/Service Costs	$0	$80	$80			
Marketing Costs	$0	$0	$0			
Customer Profit	$0	$320	$320			
Prob. of being active at end of period	100.00%	100.00%	59.00%	34.81%		
Profit expected on average	$0	$320.00	$188.80			
Present Value of Exp. Profits	$0	$320				

- Discount rate is 10% annually
- What is the churn rate? What about the retention rate?
- What assumption are we making about the timing of churn (Optimistic or Pessimistic)?
- What assumption are we making about the timing of payment (Optimistic or Pessimistic)?

TASK 6: Evaluate model performance (review slow-auc.ipynb)

CONVERT A PROBABILITY TO A BINARY OUTCOME USING BREAKEVEN AS THE THRESHOLD

		Predicted	
		Pos.	Neg.
Actual	Pos.	TP	FN
	Neg.	FP	TN

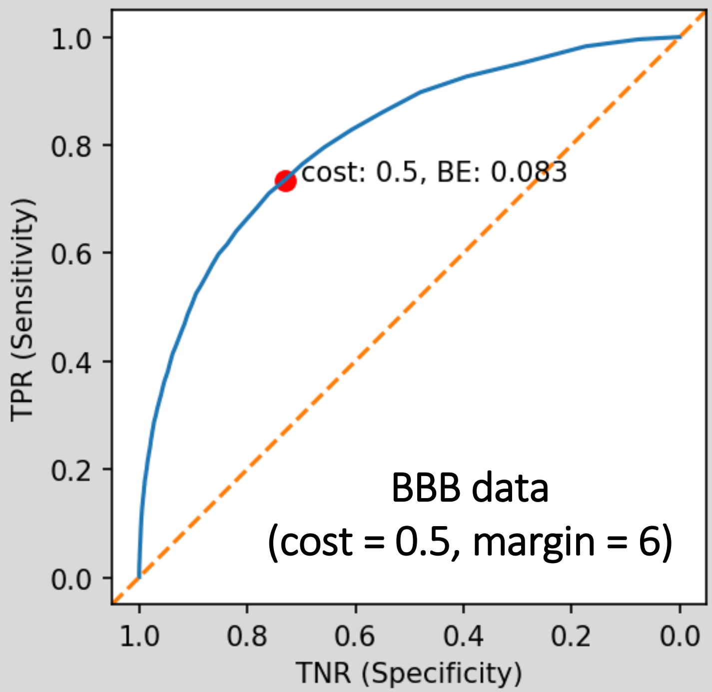
		Predicted	
		Pos.	Neg.
Actual	Pos.	655	176
	Neg.	10,871	16,176

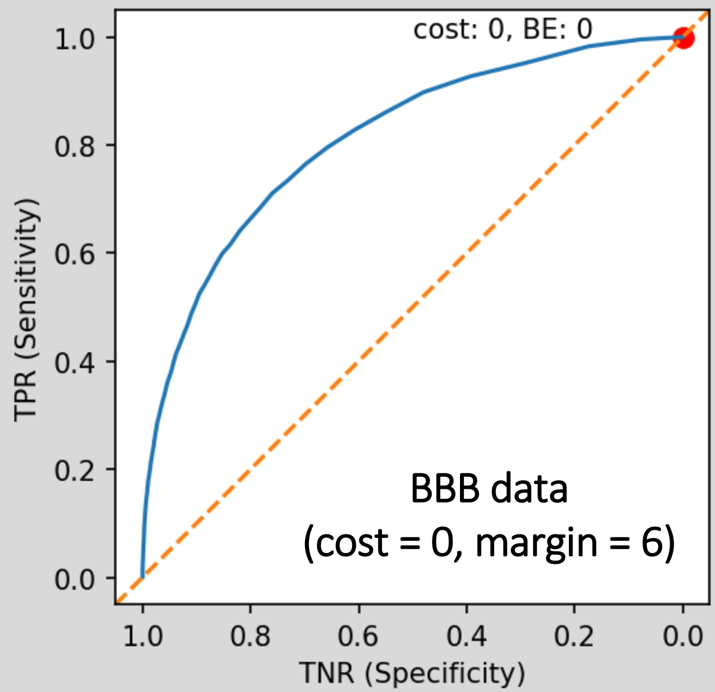
- **TP**: True positive (predicted pos, actual pos)
- **TN**: True negative (predicted neg, actual neg)
- **FP**: False positive (predicted pos, actual neg)
- **FN**: False negative (predicted neg, actual pos)

Additional performance metrics used in practice

- **Accuracy**: Proportion of all outcomes that was correctly predicted as either positive or negative, i.e., (TP + TN) / (TP + TN + FP + FN)
- **Kappa**: Corrects the accuracy measure for the probability of generating a correct prediction purely by chance
- **True positive rate (TPR)**: Proportion of **actual positive outcomes** in the data that received a **positive prediction** (i.e., TP / (TP + FN)). Also known as **sensitivity** or **recall**
- **True negative rate (TNR)**: Proportion of **actual negative outcomes** in the data that received a **negative prediction** (i.e., TN / (TN + FP)). Also known as **specificity**
- **AUC:** Area Under the (ROC) Curve. The ROC curve plots the FPR against the TPR for all possible classification thresholds. AUC is the area under this curve. The maximum AUC value is 1 and the minimum value is 0.5

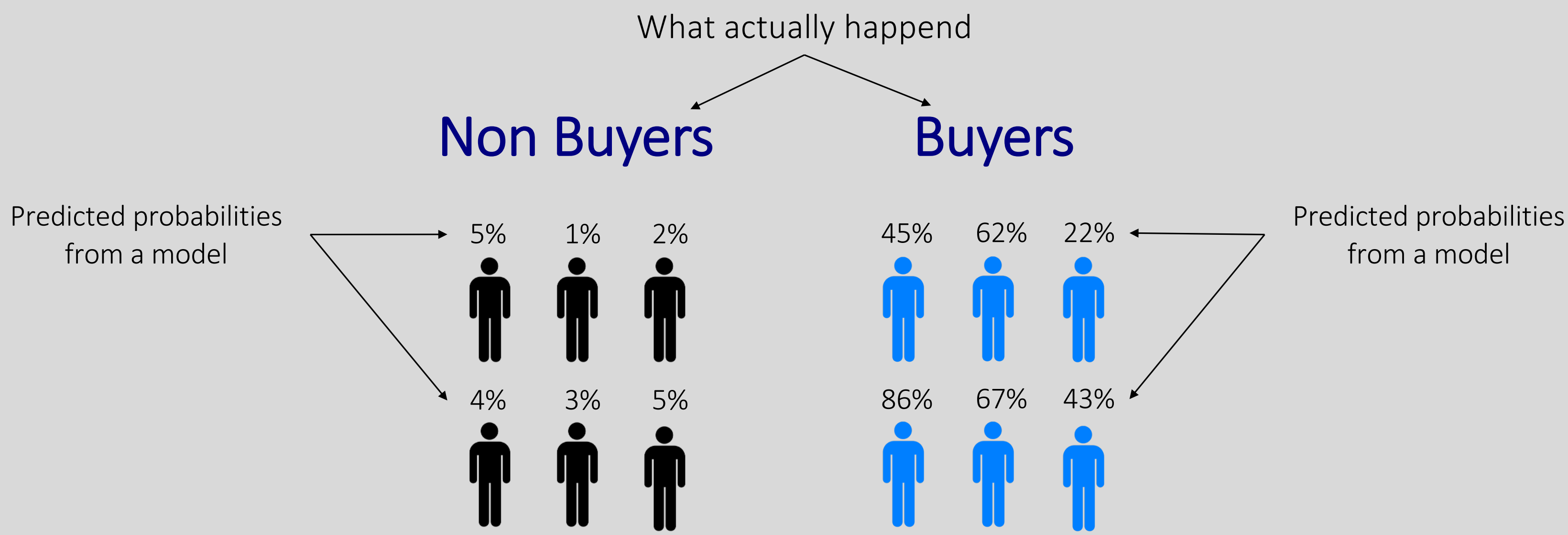
AUC is a measure of model performance at all possible thresholds





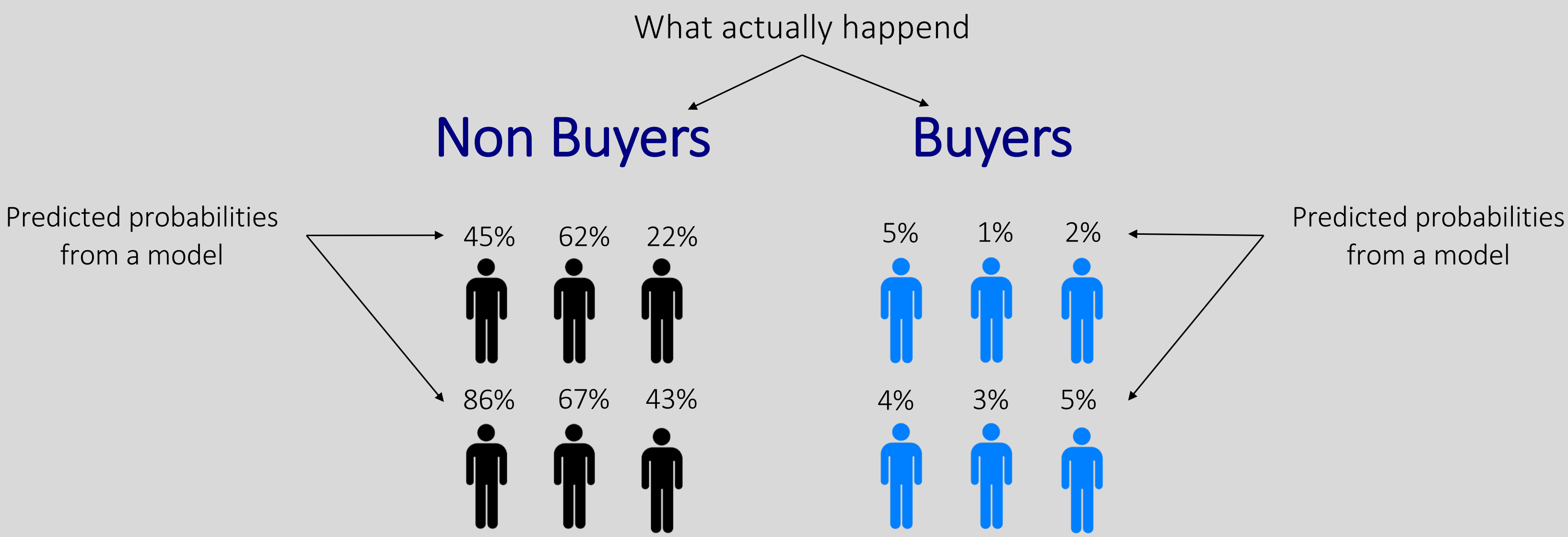
- **True positive rate (TPR)**: Proportion of **actual positive outcomes** in the data that received a **positive prediction** (i.e., TP / (TP + FN)). Also known as **sensitivity** or **recall**
- **True negative rate (TNR)**: Proportion of **actual negative outcomes** in the data that received a **negative prediction** (i.e., TN / (TN + FP)). Also known as **specificity**

Probabilistic interpretation of AUC



What does an AUC = 1 imply about “pred_did_buy” vs “pred_did_not_buy”?

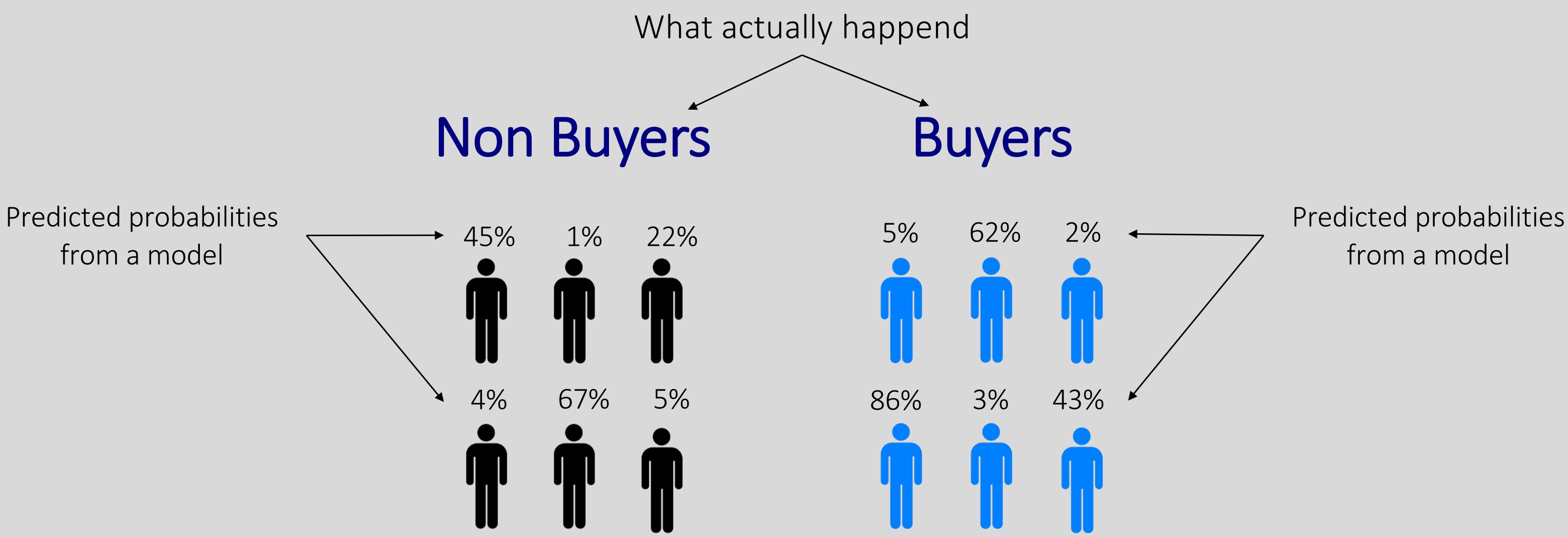
Probabilistic interpretation of AUC



What does an AUC = 0 imply about "pred_did_buy" vs "pred_did_not_buy"?

Is this a useful model?

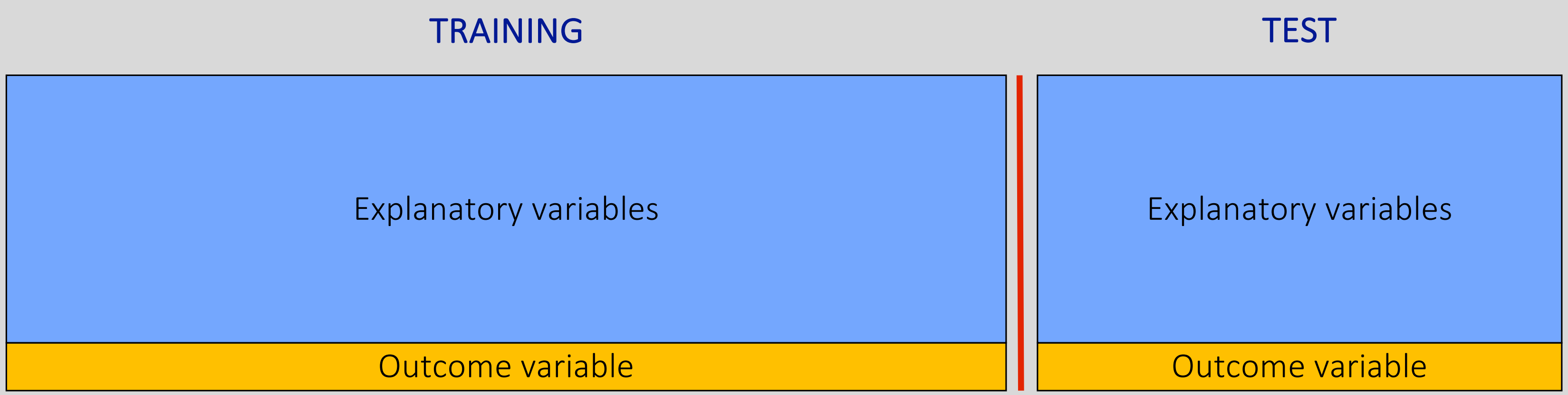
Probabilistic interpretation of AUC



What does an AUC = 0.5 imply about “pred_did_buy” vs “pred_did_not_buy”?

Is this a useful model?

TASK 7: How to “tune” hyper parameters to avoid overfitting?

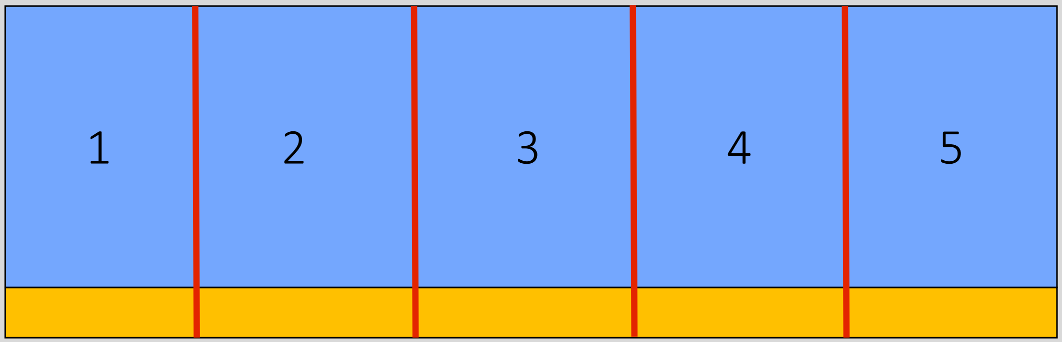


SIZE 1 2 3 4 5

DECAY 0 0.1 0.2 0.3 0.4 0.5

	Size				
Decay	1	2	3	4	5
0	(1 , 0)	(2 , 0)	(3 , 0)	(4 , 0)	(5 , 0)
0.1	(1, 0.1)	(2, 0.1)	(3, 0.1)	(4, 0.1)	(5, 0.1)
0.2	(1, 0.2)	(2, 0.2)	(3, 0.2)	(4, 0.2)	(5, 0.2)
0.3	(1, 0.3)	(2, 0.3)	(3, 0.3)	(4, 0.3)	(5, 0.3)
0.4	(1, 0.4)	(2, 0.4)	(3, 0.4)	(4, 0.4)	(5, 0.4)
0.5	(1, 0.5)	(2, 0.5)	(3, 0.5)	(4, 0.5)	(5, 0.5)

K-fold cross validation to “tune” hyper parameters



TRAIN	VALIDATE
1-4	5
2-5	1
3-1	2
4-2	3
5-3	4

HYPER PARAMETER GRID

	Size				
Decay	1	2	3	4	5
0	(1 , 0)	(2 , 0)	(3 , 0)	(4 , 0)	(5 , 0)
0.1	(1, 0.1)	(2, 0.1)	(3, 0.1)	(4, 0.1)	(5, 0.1)
0.2	(1, 0.2)	(2, 0.2)	(3, 0.2)	(4, 0.2)	(5, 0.2)
0.3	(1, 0.3)	(2, 0.3)	(3, 0.3)	(4, 0.3)	(5, 0.3)
0.4	(1, 0.4)	(2, 0.4)	(3, 0.4)	(4, 0.4)	(5, 0.4)
0.5	(1, 0.5)	(2, 0.5)	(3, 0.5)	(4, 0.5)	(5, 0.5)

The model associated with each cell in the “grid” is evaluated 5 times in a training-validation pair. The average performance metric for each grid cell is then used to determine the best hyper parameters to use.

TASK 7: K-fold cross validation to "tune" hyper parameters for NN (classification) – see bbb_sklearn.ipynb

```
nr_hnodes = range(1, 5)
hls = list(zip(nr_hnodes)) + list(zip(nr_hnodes, nr_hnodes))
hls
```

```
[(1,), (2,), (3,), (4,), (1, 1), (2, 2), (3, 3), (4, 4)]
```

```
param_grid = {"hidden_layer_sizes": hls, "alpha": [0.001, 0.01, 0.05]}
scoring = {"AUC": "roc_auc"}
```

```
clf_cv = GridSearchCV(
    clf, param_grid, scoring=scoring, cv=5, n_jobs=4, refit="AUC", verbose=5
).fit(Xs[training == 1], y[training == 1])
```

```
Fitting 5 folds for each of 24 candidates, totalling 120 fits
```

TASK 8: Experimental design and partial factorials (see bizware-review.ipynb)

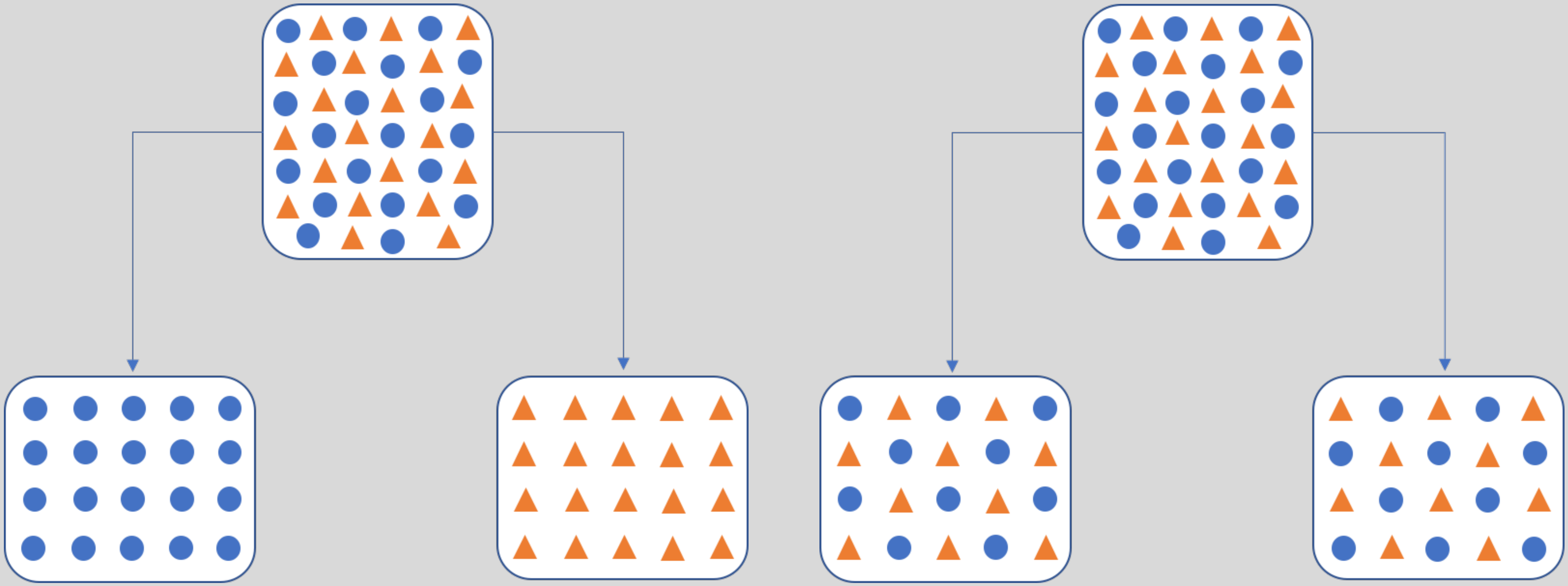
price	message	promotion	response
USD150	speed	trial	0.14
USD150	power	gift	0.40
USD160	power	trial	0.09
USD160	speed	gift	0.13
USD170	power	trial	0.06
USD170	speed	gift	0.10
USD180	speed	trial	0.01
USD180	power	gift	0.07

source: Boost your Marketing ROI with Experimental Design (HBR)
Authors: Eric Almquist and Gordon Wyner

Assume the sample size for each cell was 2,000

- Generate a partial factorial design using information about factors and levels shown in the response table (**use the radiant browser interface or functions directly from radiant.design**)
- Did you get the same design? Why (not)?
- Estimate a logistic regression based on the response table shown and predict response for all profiles
- Use **data/bizware.xls**
- What are the 2 top offers?
- What are the 2 worst offers?

Task 9: Decision trees -- Best possible split vs worst possible split



Classification trees split the data by filtering on a variable

buyer	a	b
yes	1	1
no	0	1
yes	1	1
no	0	1
yes	1	0
no	0	0
yes	1	0
no	0	0

50% buyer, 50% non-buyers

```
1  tf[tf.a == 1]
```

✓ 0.1s

	buyer	a	b
0	yes	1	1
2	yes	1	1
4	yes	1	0
6	yes	1	0

100% buyers

```
1  tf[tf.a == 0]
```

✓ 0.4s

	buyer	a	b
1	no	0	1
3	no	0	1
5	no	0	0
7	no	0	0

100% non-buyers

Classification trees split the data by filtering on a variable

buyer	a	b
yes	1	1
no	0	1
yes	1	1
no	0	1
yes	1	0
no	0	0
yes	1	0
no	0	0

50% buyer, 50% non-buyers

```
tf[tf.b == 1]
```

✓ 0.3s

	buyer	a	b
0	yes	1	1
1	no	0	1
2	yes	1	1
3	no	0	1

50% buyer, 50% non-buyers

```
tf[tf.b == 0]
```

✓ 0.2s

	buyer	a	b
4	yes	1	0
5	no	0	0
6	yes	1	0
7	no	0	0

50% buyer, 50% non-buyers

Regression trees also split the data by filtering on a variable

sales	a	b
20	1	1
10	0	1
20	1	1
10	0	1
20	1	0
10	0	0
20	1	0
10	0	0

SSE = 200

```
1  tf[tf.a == 1]
```

✓ 0.2s

	sales	a	b
0	20	1	1
2	20	1	1
4	20	1	0
6	20	1	0

SSE = 0

```
1  tf[tf.a == 0]
```

✓ 0.2s

	sales	a	b
1	10	0	1
3	10	0	1
5	10	0	0
7	10	0	0

SSE = 0

Regression trees also split the data by filtering on a variable

sales	a	b
20	1	1
10	0	1
20	1	1
10	0	1
20	1	0
10	0	0
20	1	0
10	0	0

SSE = 200

```
1  tf[tf.b == 1]
✓ 0.2s
```

	sales	a	b
0	20	1	1
1	10	0	1
2	20	1	1
3	10	0	1

SSE = 100

```
1  tf[tf.b == 0]
✓ 0.2s
```

	sales	a	b
4	20	1	0
5	10	0	0
6	20	1	0
7	10	0	0

SSE = 100

TASK 10: Calculate "node impurity" when there are two classes (CART)

$$I(A) = p_1 \times (1 - p_1) + p_2 \times (1 - p_2)$$

$$\Delta I = N(A)I(A) - N(A_L)I(A_L) - N(A_R)I(A_R)$$

$I(A)$ is the level of *impurity* in the node we want to split

$N(A)$ is the number of observations in the node we want to split

$I(A_L)$ and $I(A_R)$ represent the level of *impurity* in the node's children after the split

$N(A_L)$ and $N(A_L)$ are the number of observations in the node's children after the split

Creating a decision tree starts at the root (node)

Female variable:

- Cross tab "female" and "response"
- Calculate the reduction in impurity from the split

```
Pivot table
Data        : cart_demo50
Categorical : response female

 female yes  no Total
    yes 130  87   217
     no  71 114   185
  Total 201 201   402
```

female == “yes” vs female == “no”

- Root: 402 x (201/402 x (1-201/402) + 201/402 x (1 - 201/402))

Now evaluate all possible splits of the root node using age

Age variable:

- Cross tab "age" and "response"
- Calculate the reduction in impurity for each split

```
Pivot table
Data          : cart_demo50
Categorical : response age

   age  yes  no  Total
     1   36  71    107
     2   80  72    152
     3   85  58    143
 Total  201 201    402
```

- age == 1 vs age == 2 | age == 3

Finally, evaluate all possible splits of the root node using income

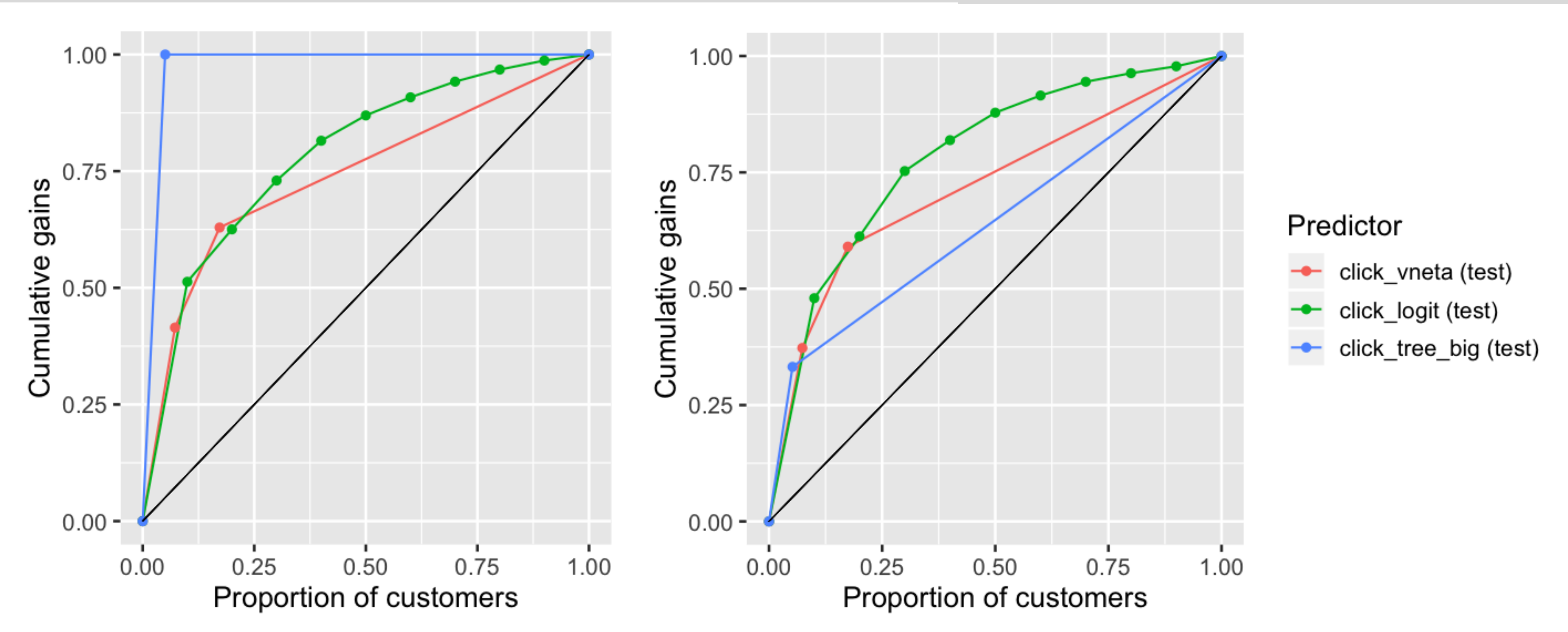
Income variable:

- Cross tab "income" and "response"
- Calculate reduction in impurity for each split

```
Pivot table
Data          : cart_demo50
Categorical : response income

 income yes  no Total
      1  42  39    81
      2  74 107   181
      3  85  55   140
  Total 201 201   402
```

- income == 2 vs income == 1 | income == 3

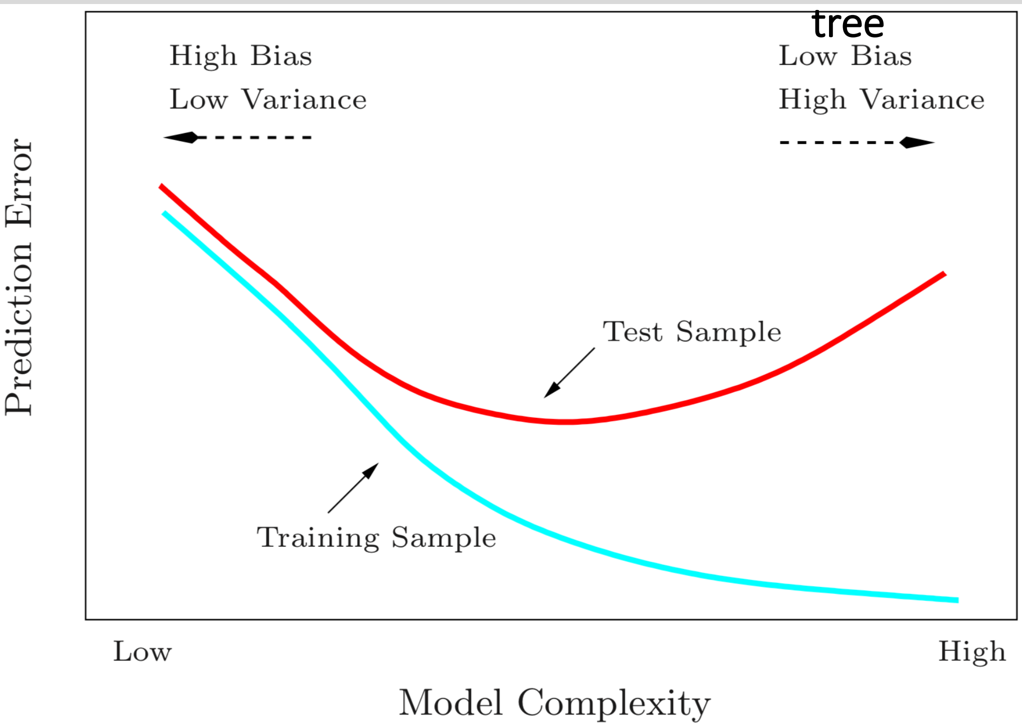
An un-pruned decision tree over-fits the training data massively!



“Ensembles” of trees address key weakness of single decision trees

Small decision tree

Large decision tree



Underfitting

Overfitting

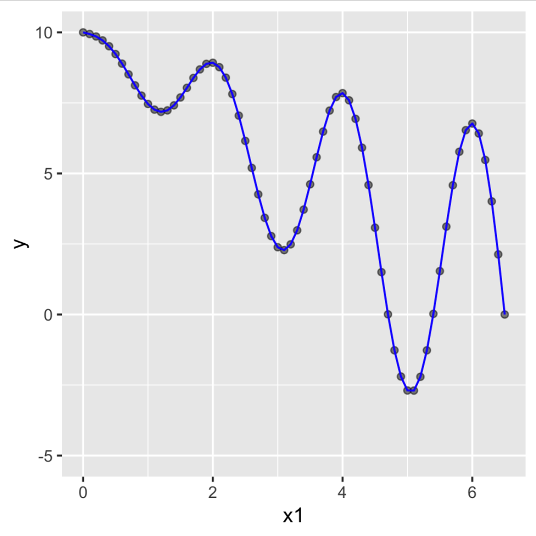
- Random Forests combine many large (overfit) decision tree to reduce variance
- Boosted Decision tree combine may small (underfit) decision trees to reduce bias
- Graph source: **The Elements of Statistical Learning**

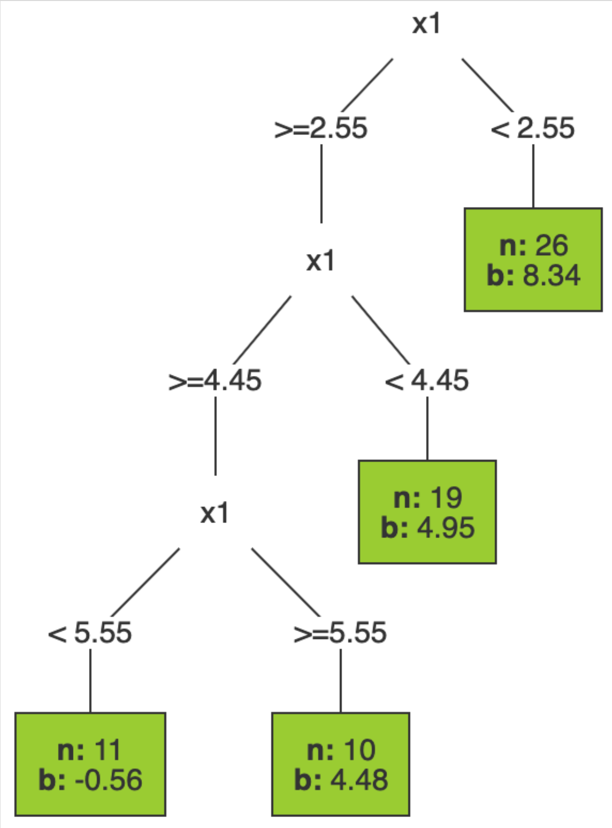
How does a random forests work?

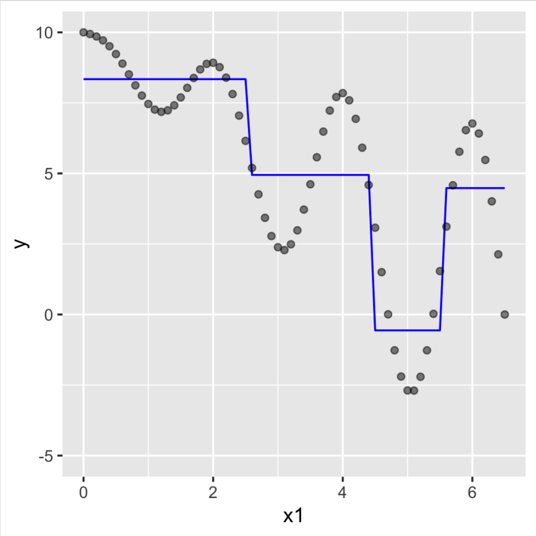
RANDOM FOREST IDEA

- Algorithm adds randomness to address overfitting for decision trees (Breiman and Cutler)
- Key idea is to create many decision trees, each of based on a
  - randomly chosen subsample of the data
  - randomly chosen subset of the explanatory variables at each node
- Can be used with different decision tree algorithms (e.g., CART)
- Very accurate predictor that can handle large numbers of explanatory variables [**WHY?**]

How do Boosted Decision Trees work?

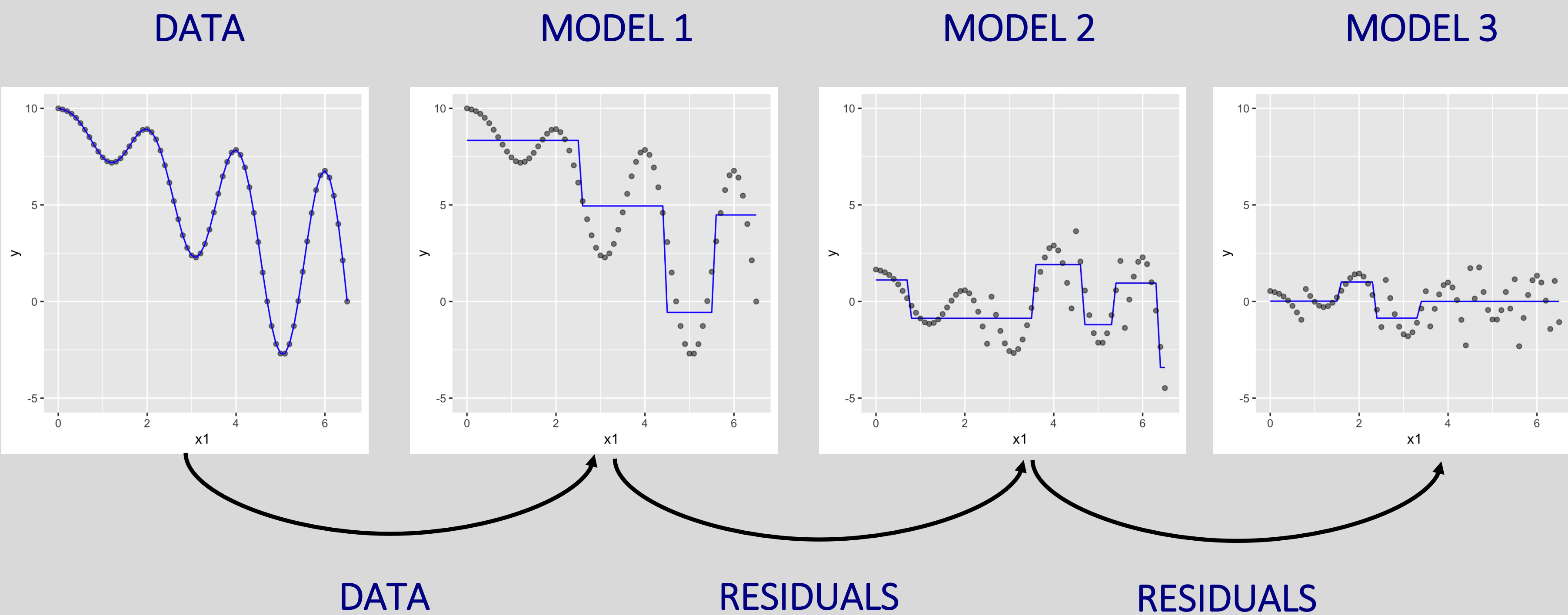




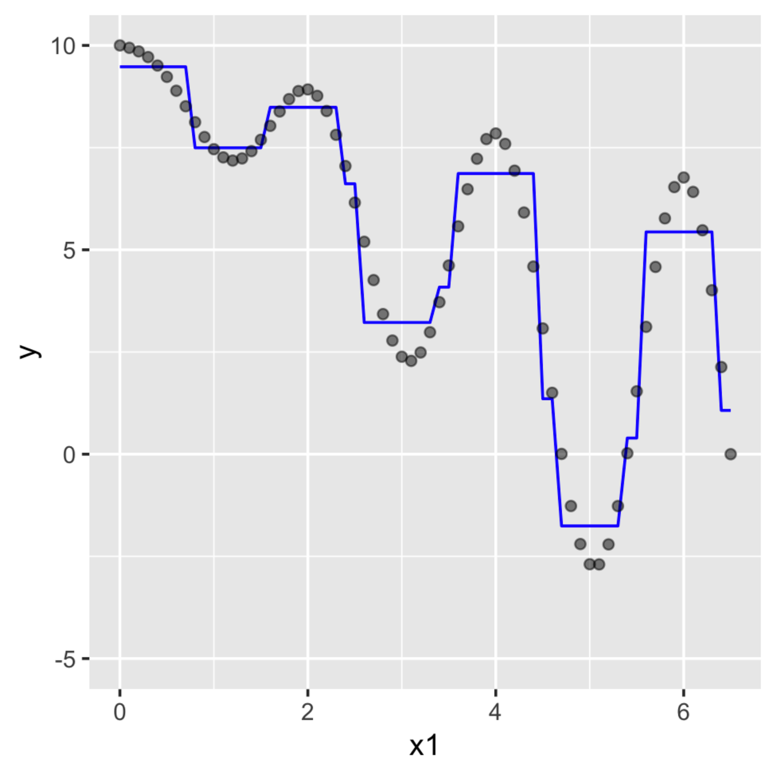


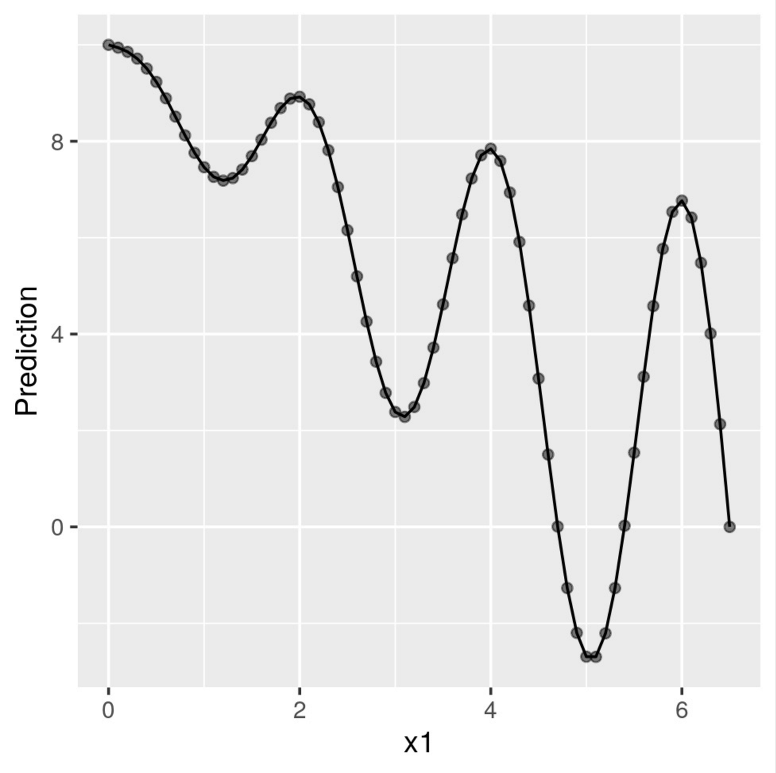
Task 10: Review code for a simplified boosted regression tree (see task-10-boosting_regression.ipynb)

Boosted Decision Trees combine “weak learners” applied to residuals from previous model(s)



To generate a final prediction, we can sum the 3 tree predictions





- Note: The above prediction uses a “learning rate” of 1. In practice, we would use a much smaller number (e.g., 0.01) and build (many) more trees