

Player Modeling: Regression and Decision Trees

Matthew Guzdial

guzdial@ualberta.ca



**UNIVERSITY
OF ALBERTA**

Announcements

- HW4 went out Monday (helper video too!)
- Practice quiz Friday
- Quiz 4 next week
 - Released a day early (Thursday at 11am, still due at Sunday at 11am)
- Next Wednesday: Initial Vote on Future of Game AI topics

Demystifying Grad School

Nov 17 5-7pm

bit.ly/cs-gradschool-ws

HOW DO I APPLY TO GRAD SCHOOL?

WHAT EXACTLY IS RESEARCH?

IS GRAD SCHOOL RIGHT FOR ME?


DEMYSTIFYING GRAD SCHOOL

Virtual Workshop
November 17 5-7pm

Open to all undergrad students!
RSVP today by scanning the code!

ALL YOUR GRAD SCHOOL QUESTIONS ANSWERED HERE!

bit.ly/cs-gradschool-ws

Hosted by  UNIVERSITY OF ALBERTA Department of Computing Science
Equity, Diversity, and Inclusion Committee

ualberta.ca/computing-science/about-the-department/edi.html

Midterm Course Feedback

(Thanks for all the positive comments!)

- "There are too many reading tasks (too fast in processing, we can totally slow it down and understand the textbook much better)" and "ThoughQuestion"
- "The instructor is helpful in clearing questions or concepts up and the course is pretty organized."
- "The assignments are very challenging, and incredibly time consuming. But they are overall enjoyable."
- "Stressful assignments (more time needed for each assignment, how about just leave with 2 or 3 assignments in total for the next time)"
- "Don't gain anything from the homework, seem like arbitrary assignments that don't reinforce info learned in class enough. Would prefer to not work on a very small part of a large codebase."
- "more details in the lecture slides would be better."
- "Maybe reserve a single lecture to learn some basic C#. E.g. how to create a list."
- "Make a way to request extensions more streamlined, or extend quiz deadline by 2 days (the weekends can be a really busy period for some people.)"
- "Consider splitting up the coding assignments into more smaller ones"
- "Prof did a good job in teaching, but it's poor in course plan and setting."

Monday: K means and K medoids clustering

Making Clusters Useful

- Predicted Based on Center: Cluster a new datapoint to the closest center, and use that center's values.
- **KNN (K-nearest neighbors):** Take the closest K (different K than K means) datapoints to this new datapoint, and take the majority value(s)

KNN Example on (Virtual?) White Board

Most Commonly Desired Prediction? Churn

- Churn rate: The rate at which customers cut ties with a company
- In games this is how quickly a game loses players. After how long do they stop playing?
- Churn is one of the most common problems player analytics teams are tasked with

Regression

- If we can map variables onto likelihood of churn (how long the players will play) we can figure out what the designers need to fix!
- Two common approaches
 - Linear/logistic regression
 - Decision trees

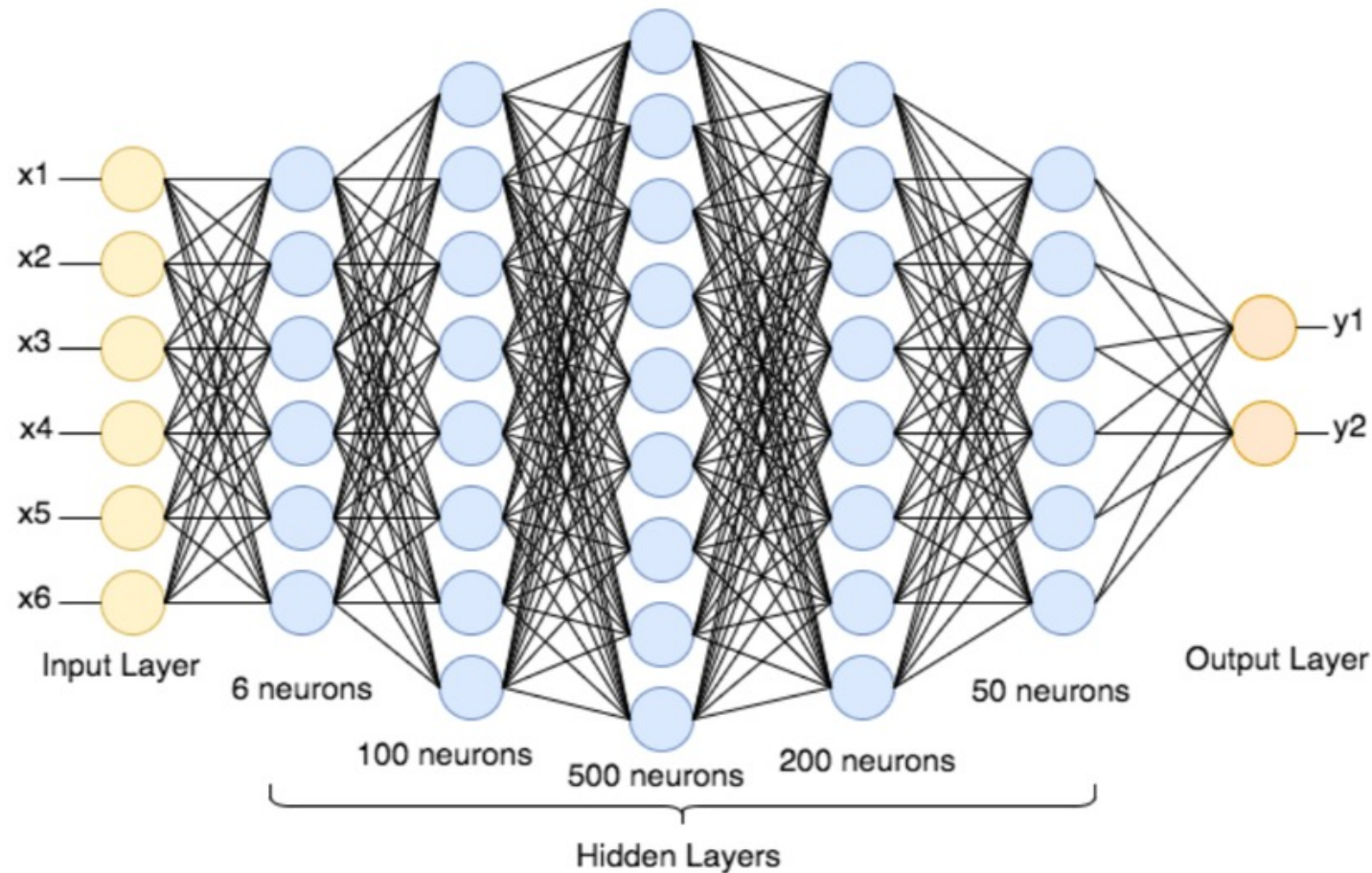
Only two regression techniques?

- No of course not.
- But similar story as with K means...

Algorithm	Level 1	Levels 1 and 2
Logistic regression	48.3	77.3
MLP/Backpropagation	47.7	70.2
J48 (C4.5) decision tree (pruned)	48.7	77.4
REPTree decision tree (pruned)	48.5	77.2
Multinomial naive bayes	43.9	50.2
Bayes network	46.7	65.1
SMO Support vector machine	45.9	70.0
Baseline	39.8	45.3

Why not Deep Neural Networks?

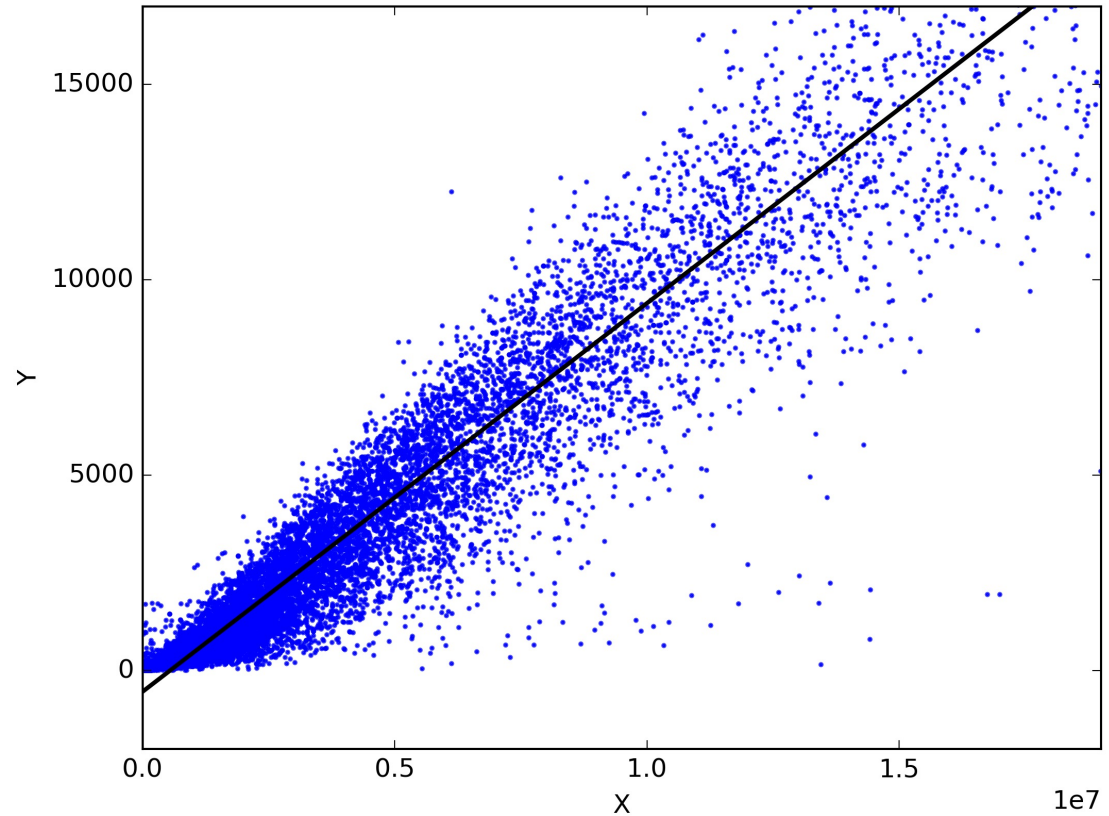
We need *interpretable* results (and extra work)



Linear Regression

Given input X and expected output Y , find m and b such that:

$$Y = m * X + b$$



Linear Regression Algorithm

Find the best fit values for m (slope) and b (y-intercept)

Lots of algorithms to do this

- E.g. Minimizing the sum of squared residuals

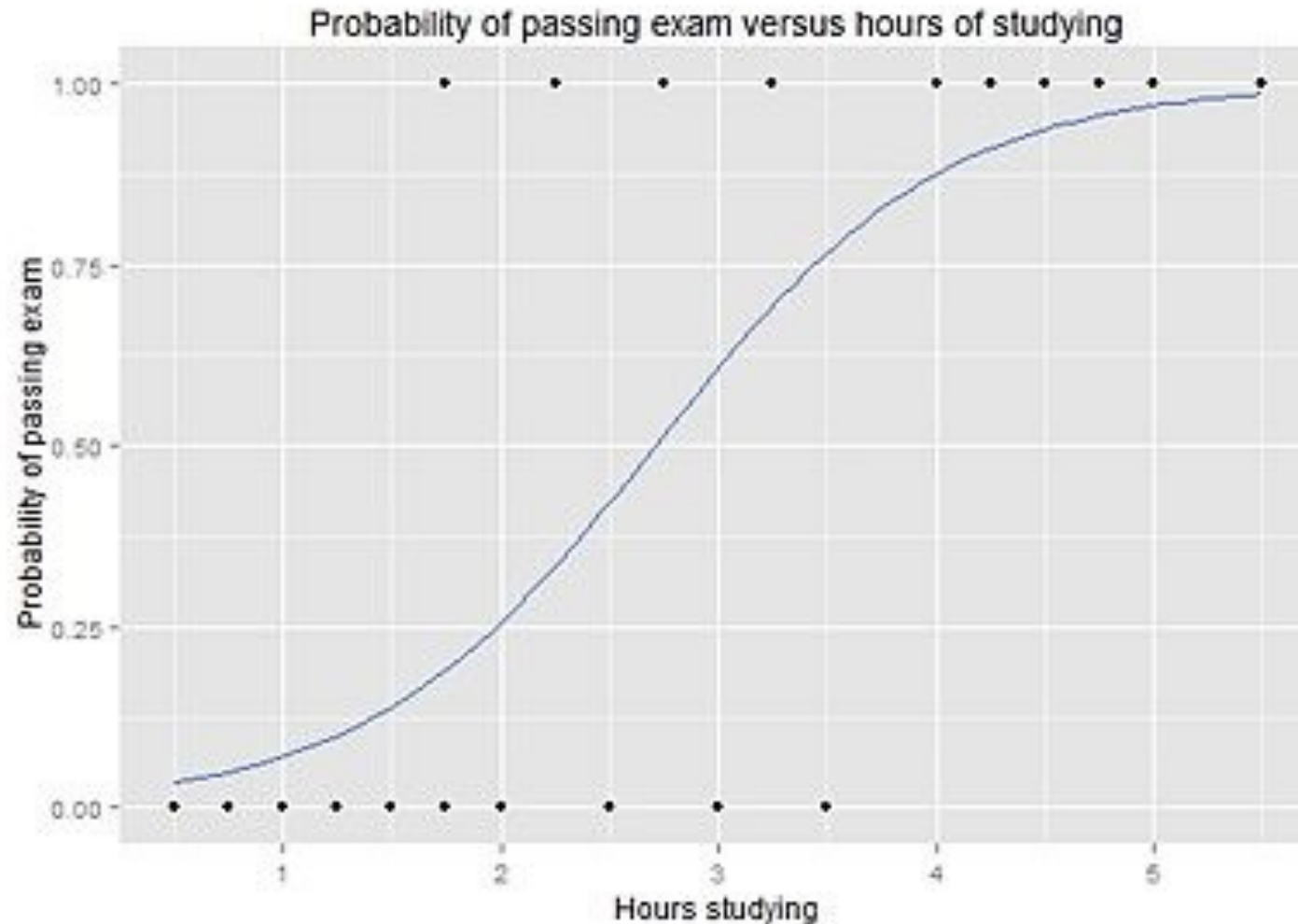
$$\text{Find } \min_{\alpha, \beta} Q(\alpha, \beta), \quad \text{for } Q(\alpha, \beta) = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - \alpha - \beta x_i)^2$$

We won't worry about *how* we get the values here

Logistic Regression

(Same idea but fit data with logistic function)

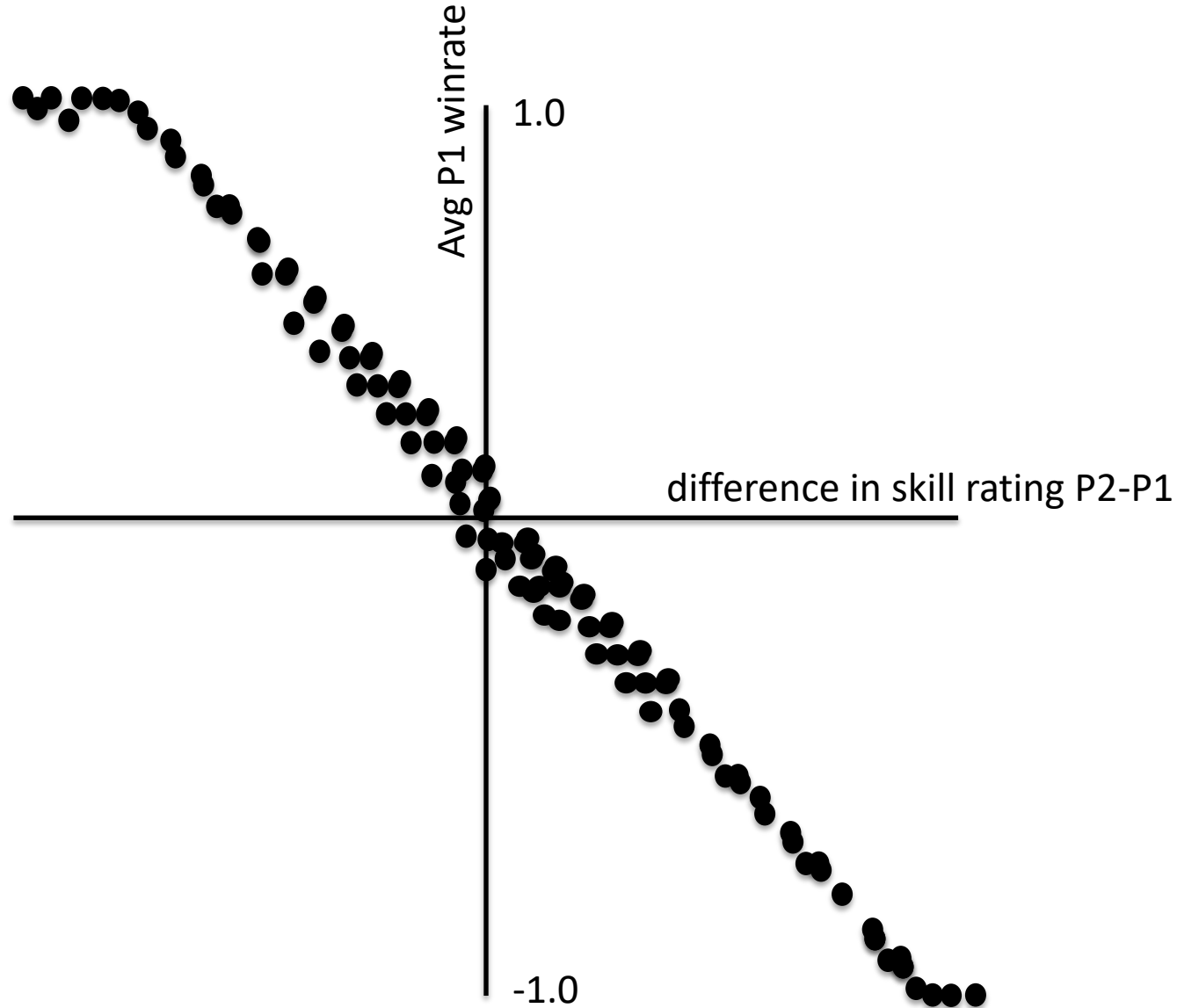
$$\text{logistic}(\eta) = 1 / (1 + \exp(-\eta))$$



PQ1: Regression <https://forms.gle/4gS6qADhB79v6Kpn9>
<https://tinyurl.com/guz-pq23a>

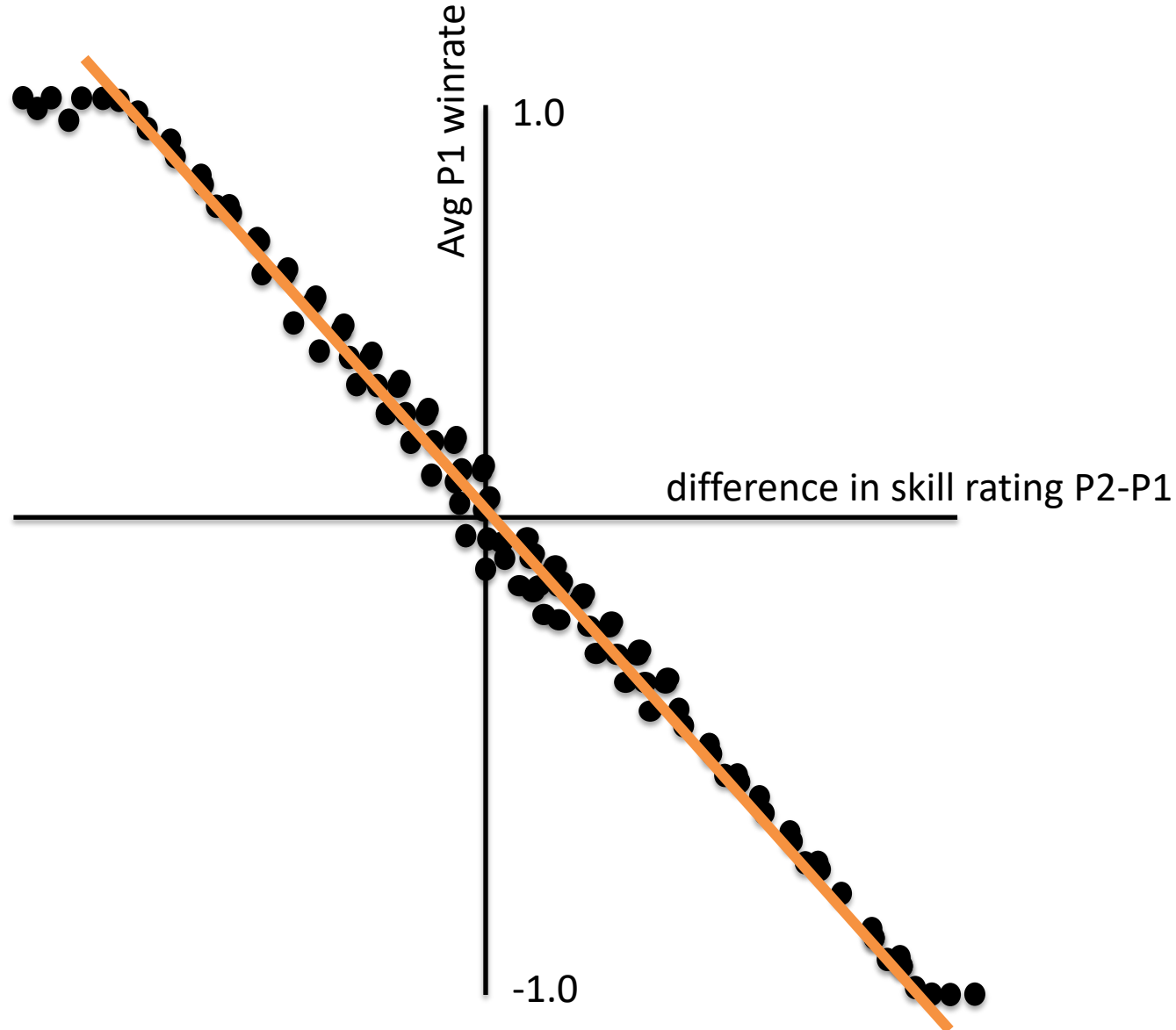
Given the dataset on the right,
would you use a linear
regression or logistic
regression?

Why?



My Answer: Linear!

The data “looks” logistic (if mirrored), but that’s just because average winrate caps out at +1.0 and -1.0



Regression

- Pros:
 - Fast to build
 - Fast to use
 - Simple but powerful
- Cons:
 - Too simple for complex mappings
 - Overly dependent on training data
 - Can't work with partial information
 - Errors can massively harm its performance

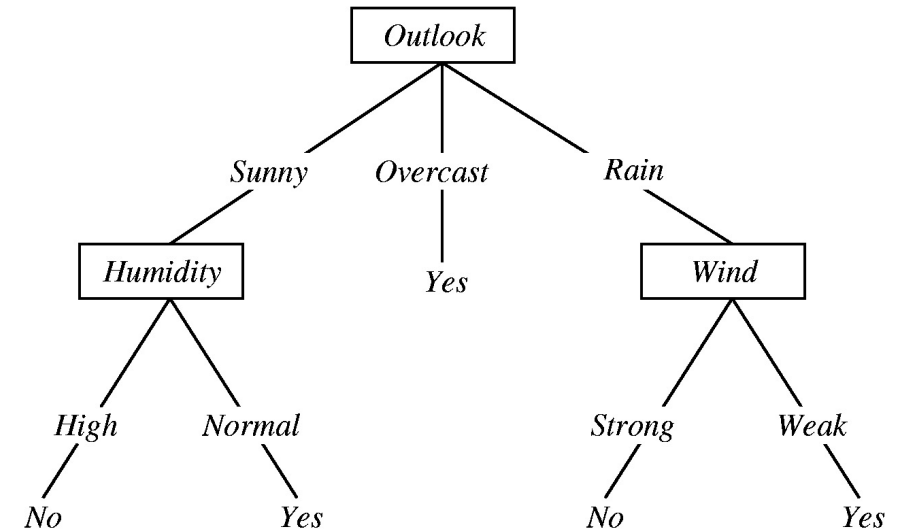
What if we want to learn a more complicated function? (Still interpretable/easy to run)

Decision Trees

Learn a tree to map input (X) to output (Y) from training data.

Choice Node: Some check (is var > some threshold, Boolean variable check, etc.)

Decision Node: Final output.



Decision Tree Algorithm

```
set currentNode to head
  add all training data to head
openNodes = [currentNode]
while currentNode.entropy*>threshold:
  currentNode = openNodes.pop()
  children = calculate best split point(s)
  for child in children:
    if child.entropy<=threshold:
      make decision node
    else:
      openNodes.push(child)
return head
```

*entropy is the measure of how much in agreement the data “in” this node is.

Basically, we want to minimize: $1 - (\text{number of majority class in node}) / (\text{number of datapoints in node})$

Decision Tree Example on (Virtual?) White Board

Decision Trees are suited to problems where...

- Data is in the form X,Y pairs
- Y has discrete values it can be (yes/no or in a category)
 - (Can be applied to regression instead of classification problems)
- $X \rightarrow Y$ is not a simple linear/logistic function
- Training data may contain errors/noise
- Training data may contain some X values without all attributes

Decision Trees

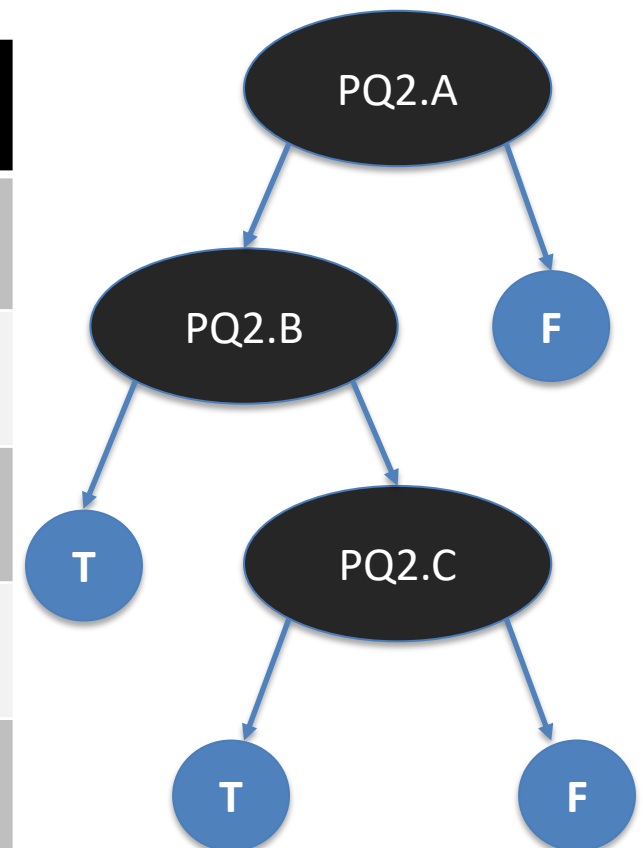
- Pros:
 - Fast to use
 - Robust to errors
- Cons:
 - Slow to build
 - Depending on threshold for entropy, can **overfit** to training data

PQ2: Decision Trees <https://forms.gle/KSVBY3VZndrnTTV37> <https://tinyurl.com/guz-pq23b>

Give me the decision tree (based on answering questions about it) for the following data. Assume threshold of 0 (perfect agreement).

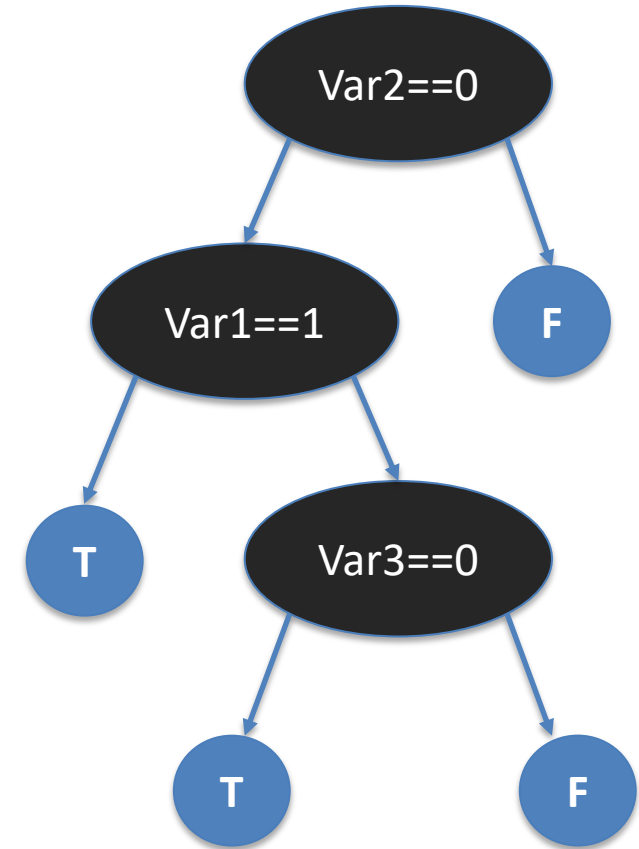
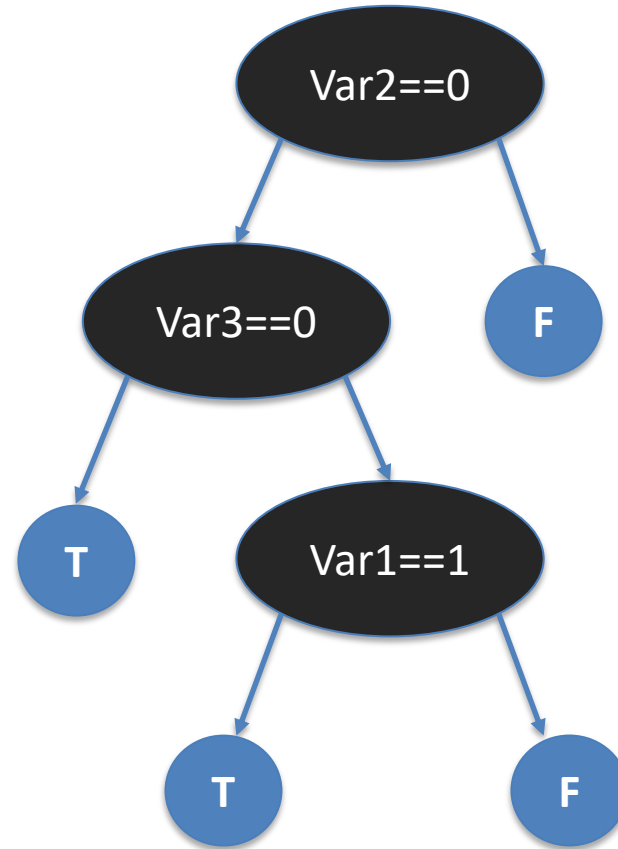
Minimize: $1 - (\text{number of majority class in node}) / (\text{number of datapoints in node})$

id	Var1	Var2	Var3	Churned?
1	1	0	1	T
2	1	1	1	F
3	0	0	0	T
4	0	0	1	F
5	1	0	0	T



Answers

id	Var1	Var2	Var3	C?
1	1	0	1	T
2	1	1	1	F
3	0	0	0	T
4	0	0	1	F
5	1	0	0	T

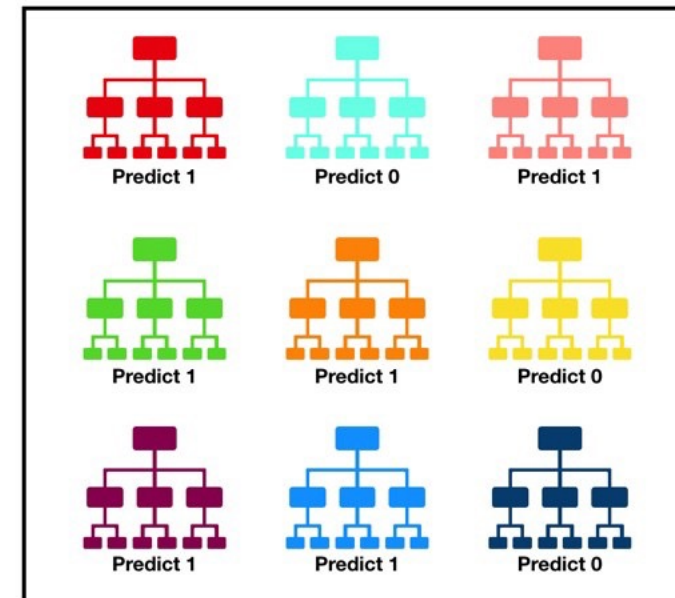


Overfitting

- In which a model gets *too* good at predicting training data at the risk of making mistakes during testing.
- Child sees three dogs, all of different breeds, and is told that they are each a different breed but all dogs
- Child sees a cat and asks what breed of dog it is

Overfitting? Maybe try Random Forest

- A random forest makes use of a designer-specified number of decision trees, each trained on a random subset of the training data.
- During testing, each tree of the random forest votes, with the majority (or average or median) answer taken.



Tally: Six 1s and Three 0s
Prediction: 1

Clustering Example (time permitting)

Data in form: <death location, enemy that killed, and enemy attacked that killed>:

P1: <(100, 100, 100), boss1, boss1_attack1>

P2: <(200, 200, 200), boss1, boss1_attack2>

P3: <(220, 220, 220), boss2, boss2_attack1>

P4: <(100, 220, 100), boss2, boss2_attack2>

...

Next Up...

- Friday: Jumping back to unsupervised learning and clustering
- What if we can't take all the time to find the optimal K?
- What if we don't think every datapoint x belongs to a *single* cluster/group?
- Monday: Example of real Data Science problems in industry
- Wednesday: Voting on Future of Game AI + PCG Intro!