

Pokerbots 2025

Lecture 5: Machine Learning

Sponsors



hudson river trading



CITADEL | CITADEL Securities





Announcements

reference-4-2025

- Coding demo from yesterday available on GitHub
- Jupyter, Numpy, Matplotlib
- Recording link in readme

Make-up Progress Reports due Tonight

- As per syllabus, if your team missed the Week 1 bot submission deadline, a progress report is required to earn credit
- Reports are half a page and must list team name and members
- Submit via email to pokerbots@mit.edu by 11:59pm Tonight

Hackathon Tomorrow!

- Sponsored by **Codeium**
- Wednesday night (1/15)
- 32-044, 7pm -- LATE
- Show up and grind
- Dinner provided
- Snacks, fun, and games
- Prizes for challenges and those that stick around
- Details will be announced on Piazza

No Class Thursday

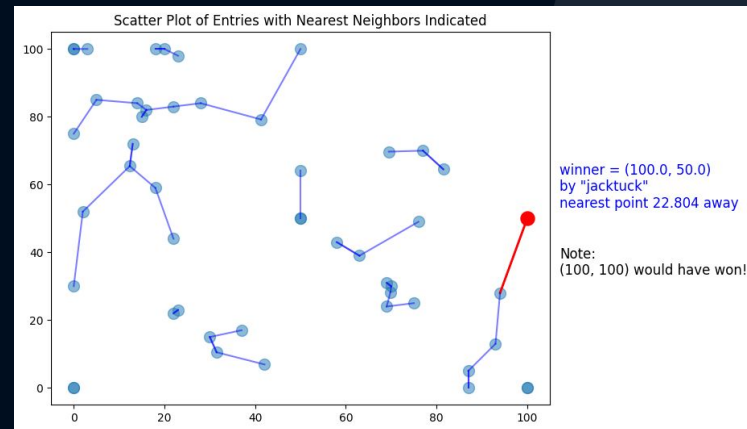
Cancelled in light of Hackathon the night before and Codeium Tech Talk at 11:30am



Giveaways!

Distance Game: pkr.bot/dist

- Pick a point (x, y) , x and y are reals from 0-100
- Winner is whoever's the furthest from anyone else's guess (maximum distance to nearest neighbor)
- Prize: Nintendo Switch



from last year^

*Banned kerbs that participated in yesterday's wager are ineligible to win

Survey Raffle: pkr.bot/survey

Class is halfway through (sort of)... help us improve it!

Winner selected at random

Prize:
One year's worth of GTO
Wizard Full Subscription

***EVERYONE** eligible to win

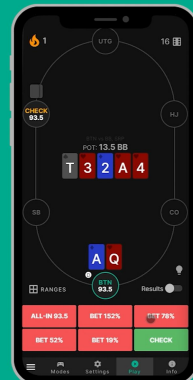


10,000,000+ presolved solutions in your pocket.

Instantly access any spot you can imagine.
From preflop to any river you want, we have all the possible situations.
It works seamlessly on your desktop and mobile.

6max Cash Spins
HU SNGs MTTs
SOON PLANNING
Your solutions PLO

[Get solutions for free](#)



The background is a solid blue color. On the left side, there are several overlapping circles of varying shades of blue, creating a layered, abstract effect. The circles are semi-transparent, allowing the blue of the background and other circles to show through.

Machine Learning

Agenda for Today

- Machine Learning Basics
- Problem Setting for Supervised Learning
- Foundational Algorithms
- Creating a Good Model



Basics

What is Machine Learning (ML)?

- Using a machine (computation) to find patterns in data
- Found patterns can be applied without explicit programming
- Key idea: *generalize* instead of *memorize*
- Typically utilize large amounts of data
- Used for prediction, data compression, automation, decision-making

ML Workflow Overview

- Acquire dataset
- Create model(s)
- Evaluate model(s)
- Hopefully have good model via selection and improvement
- Deploy!

Types of Learning

- Supervised Learning
 - Dataset has examples of correct behavior
 - “Questions with answers”
 - Prediction tasks
- Unsupervised Learning
 - Dataset has no notion of good behavior
 - Task is to find patterns to represent dataset
 - Clustering, compression, generative modeling
- Reinforcement Learning (RL)
 - Environment with feedback (rewards), but no correct target
 - Sequential decision making, control, pokerbots!

Questions?



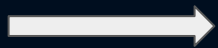
Problem Setting

Example

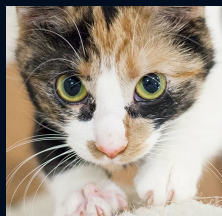
- Supervised learning setting (which we will focus on today)
- We love cats and birds!
- We want our algorithm to take a photo and tell us if it's a cat or a bird
- We have example photos and labels of both

Example

Data



"Cat"



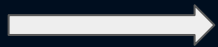
"Cat"



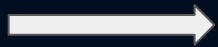
"Cat"

Labels

Example



“Bird”



“Bird”



“Bird”

Example



Cat or Bird?

Key Variables in Supervised Learning

- data example contains
 - “input” properties called *features*
 - “output” property called *label*
- *dataset* is a collection of these examples
- an *algorithm* that takes as an input an incomplete example with only features, and outputs a label to complete the example.

Types of Prediction

Classification

- Predict a category
- Example: “cat or bird”

Regression:

- Predict continuous value aka real number(s)
- Example: giveaway estimations

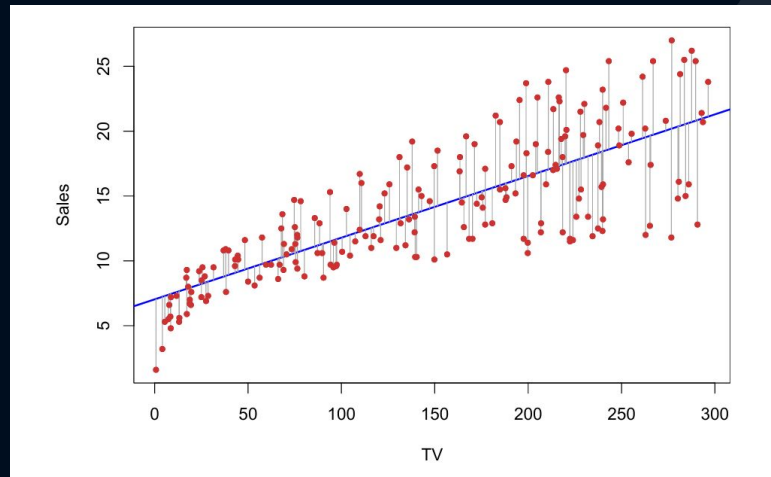
Questions?

The background is a solid dark blue. On the left side, there are several overlapping, semi-transparent shapes in a lighter shade of blue. These shapes include a large triangle pointing upwards and a large circle, creating a layered, abstract effect.

Foundational Algorithms

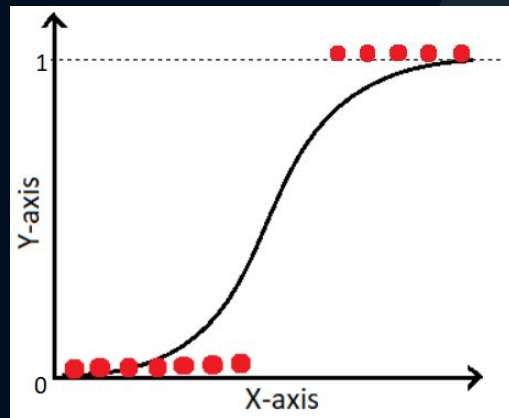
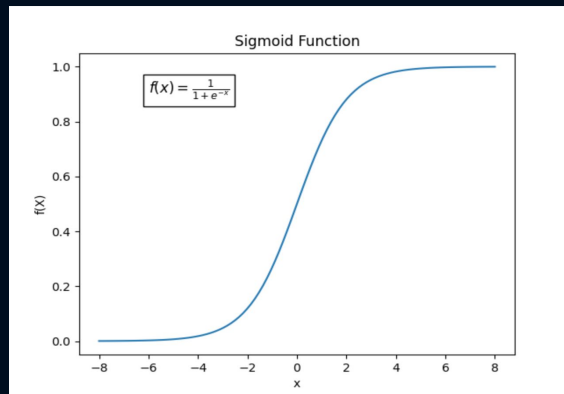
Linear Regression

- Linear relation between input and output
- “Line of best fit”
- $y = w \cdot x + b$
- Can generalize to multiple variables
 - $y = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + b$
- Easy to work with but only works well under certain conditions



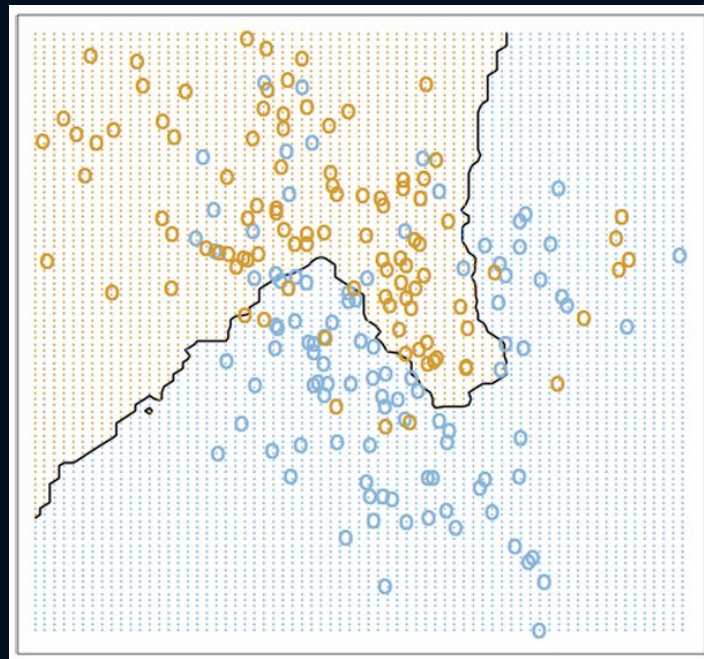
Logistic Regression

- Modifies linear regression for binary classification
- Sigmoid function $\sigma(x)$ transforms any real number onto interval (0,1).
- $y = \sigma(w \cdot x + b)$
- Interpreted as outputting a probability of belonging to a specified class



k-Nearest Neighbors (kNN)

- Classification or regression
- Decision using proximity
- For an input:
 - Find the k closest neighbors within dataset
 - Aggregate their labels to decide output
 - Majority vote for classification
 - Average for regression
- Very flexible and can represent complex patterns
- Struggles with higher dimensions



Questions?

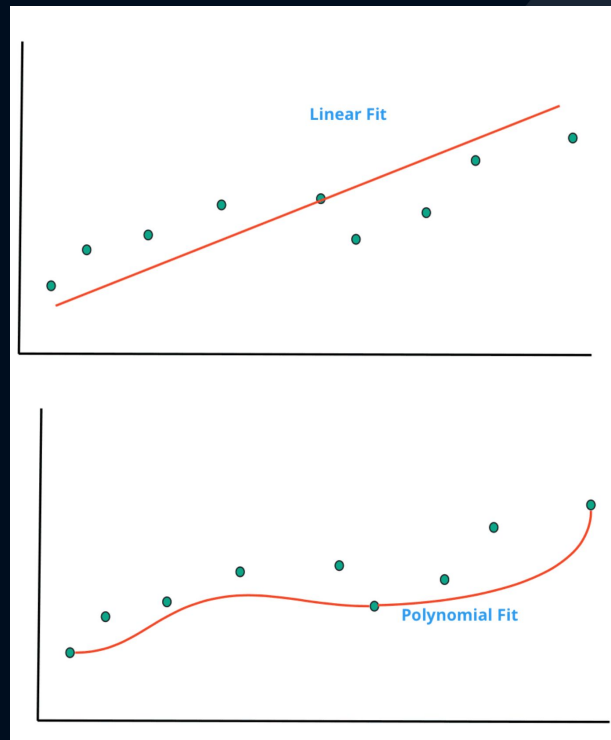


Creating a Good Model

Creating a Good Model

What is a model?

- Template for an algorithm
- *Model class* or *hypothesis class* is the collection of all possible algorithms that follow that template
- Some models are defined by *parameters*
 - Ex: w and b in linear regression
 - Space of possible parameters \leftrightarrow model class
 - Choosing a model equates to choosing the parameters
- Other models are *non-parametric* (kNN)
- The larger the model class, the more *expressive* a model is





Creating a Good Model

Loss Functions

- Metric for seeing if model is good by measuring the error it makes on an example
- This equates to measuring some distance between predicted and true label
 - Linear Regression: squared distance
 - Logistic Regression: cross-entropy
 - kNN: 0/1 accuracy rate
- Loss function reports a penalty for a single input-output example - model is evaluated based on cumulative or average loss across whole set of examples

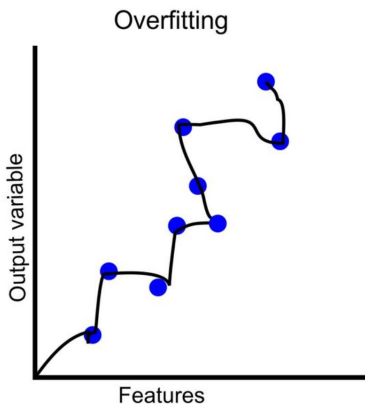
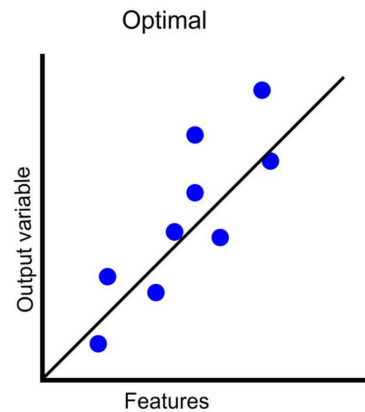
Creating a Good Model

Combine everything together!

- Consider all models within the hypothesis class and pick the one that has the best (lowest) loss across the entire dataset
- For parametric models, this is a minimization problem over the parameter space
 - “Training”
 - Sometimes near impossible to find the absolute best one so approximations are settled for
- Why don't we simply just use the most expressive hypothesis class? More options means that the best options are going to have better results...
 - Ex: the space of polynomial functions can do more than everything linear functions can

Overfitting

- “Too good to be true”
- Model fits dataset extremely well but fails to capture underlying relationship (which is what we want)
- Memorization occurs, which prevents generalization
- Caused by overexpressive hypothesis class, aka too many parameters, relative to number of data points



Model Evaluation

- Split dataset into training and testing groups
 - Create model only using train set
 - Evaluate model based on loss across test set
- Goal is to detect overfitting and have a more realistic measure of a good model
- Perform the split randomly
 - Each data point has equal chance of showing up in either
 - Idea is to have the data be from the same distribution but still different
- Analogy: Practice exams vs Real exam
 - Test same capabilities using different questions
 - If real exam was exact copy of practice, then doing well doesn't say much about what was learned
- When we deploy model it may fail - we want to assess whether this would happen (ideally not)

Sources of Error

Bias

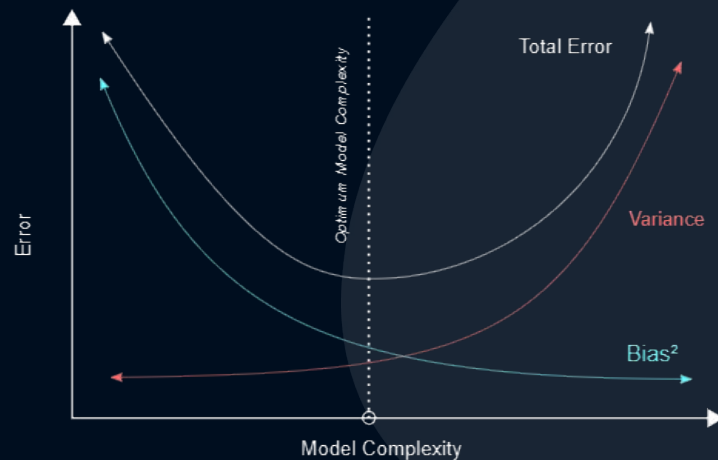
- error due to simplistic assumptions
- model class not expressive/flexible enough to represent data
- consistent error resulting from approach
- Characterized by high loss in both training and testing
- Ex: Linear Regression

Variance

- error from sensitivity to training data
- Model class is overly flexible, attempting to capture noise as a pattern
- Characterized by low loss in training but high loss in testing (aka overfitting)
- Ex: kNN

Bias-Variance Tradeoff

- There is an inherent tradeoff between bias and variance within model selection
- Many design decisions (aka hyperparameters) increase/decrease the model's complexity which could lead to too much bias or variance
- Some considerations to find the “sweet spot”:
 - Validation sets
 - Regularization
 - Careful feature selection/pruning



Questions?

Lunch 🤪

Leave any type of feedback at pkr.bot/feedback!



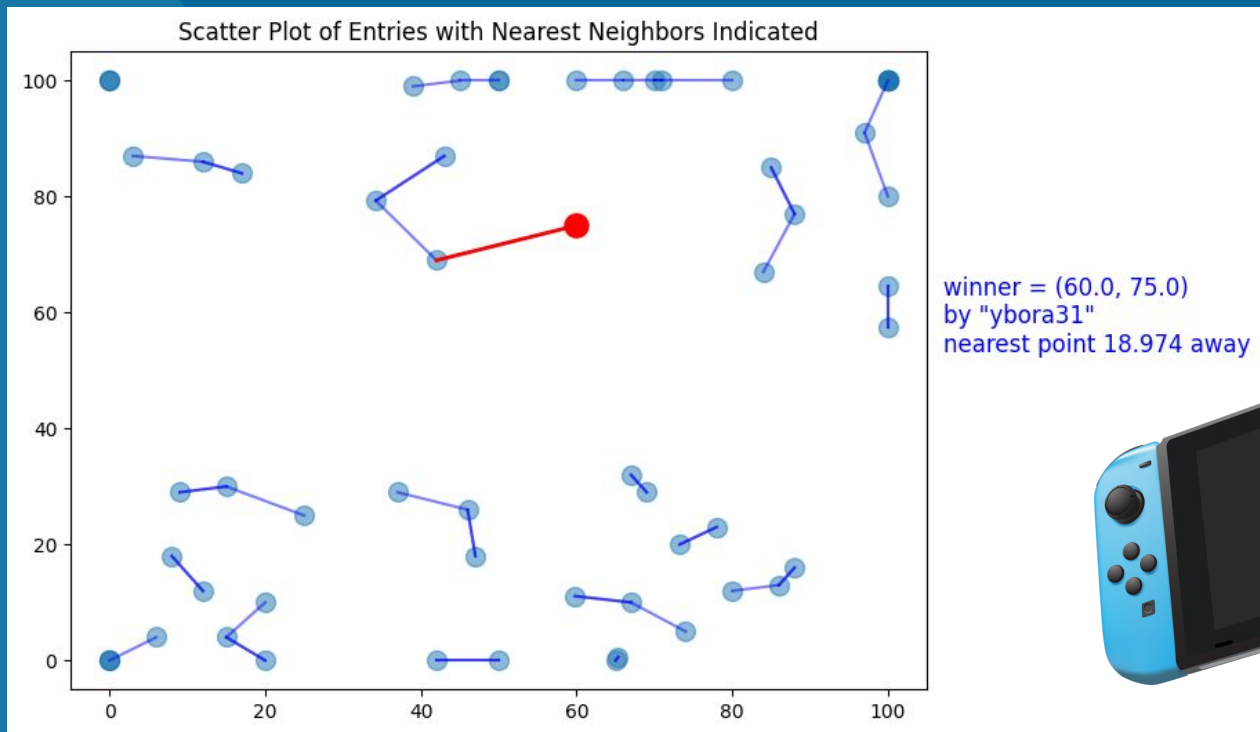


Live Coding: reference-5-2025

Survey Raffle: kerb “rlsalas”



Distance Game: kerb "ybora31"



Thanks for watching!

Slides/notes will be posted on pkr.bot/resources

Make sure to check pkr.bot/piazza for updates

Lecture recordings at pkr.bot/panopto

Leave feedback at pkr.bot/feedback!