

# Machine Learning - Classification

Professor Widom's Instructional Odyssey

[www.professorwidom.org](http://www.professorwidom.org)



# Data Tools and Techniques

- Basic Data Manipulation and Analysis  
Performing well-defined computations or asking well-defined questions (“queries”)
- Data Mining  
Looking for patterns in data
- Machine Learning  
Using data to build models and make predictions
- Data Visualization  
Graphical depiction of data
- Data Collection and Preparation

# Regression

- Supervised machine learning
- Training data, each example:
  - Set of predictor values - “independent variables”
  - Numerical output value - “dependent variable”
- Model is function from predictors to output
  - Use model to predict output value for new predictor values
- Example
  - Predictors: mother height, father height, current age
  - Output: height

# Classification

- Supervised machine learning
- Training data, each example:
  - Set of feature values - numeric or categorical
  - Categorical output value - “label”
- Model is method from feature values to label
  - Use model to predict label for new feature values
- Example
  - Feature values: age, gender, income, profession
  - Label: buyer, non-buyer

# Other Examples

## Medical diagnosis

- **Feature values:** age, gender, history, symptom1-severity, symptom2-severity, test-result1, test-result2
- **Label:** disease

## Email spam detection

- **Feature values:** sender-domain, length, #images, keyword<sub>1</sub>, keyword<sub>2</sub>, ..., keyword<sub>n</sub>
- **Label:** spam or not-spam

## Credit card fraud detection

- **Feature values:** user, location, item, price
- **Label:** fraud or okay

# Algorithms for Classification

Despite similarity of problem statement to regression, non-numerical nature of classification leads to completely different approaches

- K-nearest neighbors
- Decision trees
- Naïve Bayes
- Deep neural networks
- ... and others

# K-Nearest Neighbors (KNN)

For any pair of data items  $i_1$  and  $i_2$ , from their feature values compute  $distance(i_1, i_2)$

Example:

Features - gender, profession, age, income, postal-code

person<sub>1</sub> = (male, teacher, 47, \$25K, 94305)

person<sub>2</sub> = (female, teacher, 43, \$28K, 94309)

$distance(\text{person}_1, \text{person}_2)$

$distance()$  can be defined as inverse of  $similarity()$



# K-Nearest Neighbors (KNN)

Features - gender, profession, age, income, postal-code

person<sub>1</sub> = (male, teacher, 47, \$25K, 94305)

person<sub>2</sub> = (female, teacher, 43, \$28K, 94309)

Remember training data has labels



# K-Nearest Neighbors (KNN)

Features - gender, profession, age, income, postal-code

person<sub>1</sub> = (male, teacher, 47, \$25K, 94305) buyer

person<sub>2</sub> = (female, teacher, 43, \$28K, 94309) non-buyer

Remember training data has labels

To classify a new item  $i$ : In the labeled data find the  $K$  closest items to  $i$ , assign most frequent label

person<sub>3</sub> = (female, doctor, 40, \$40K, 95123)

# KNN Example

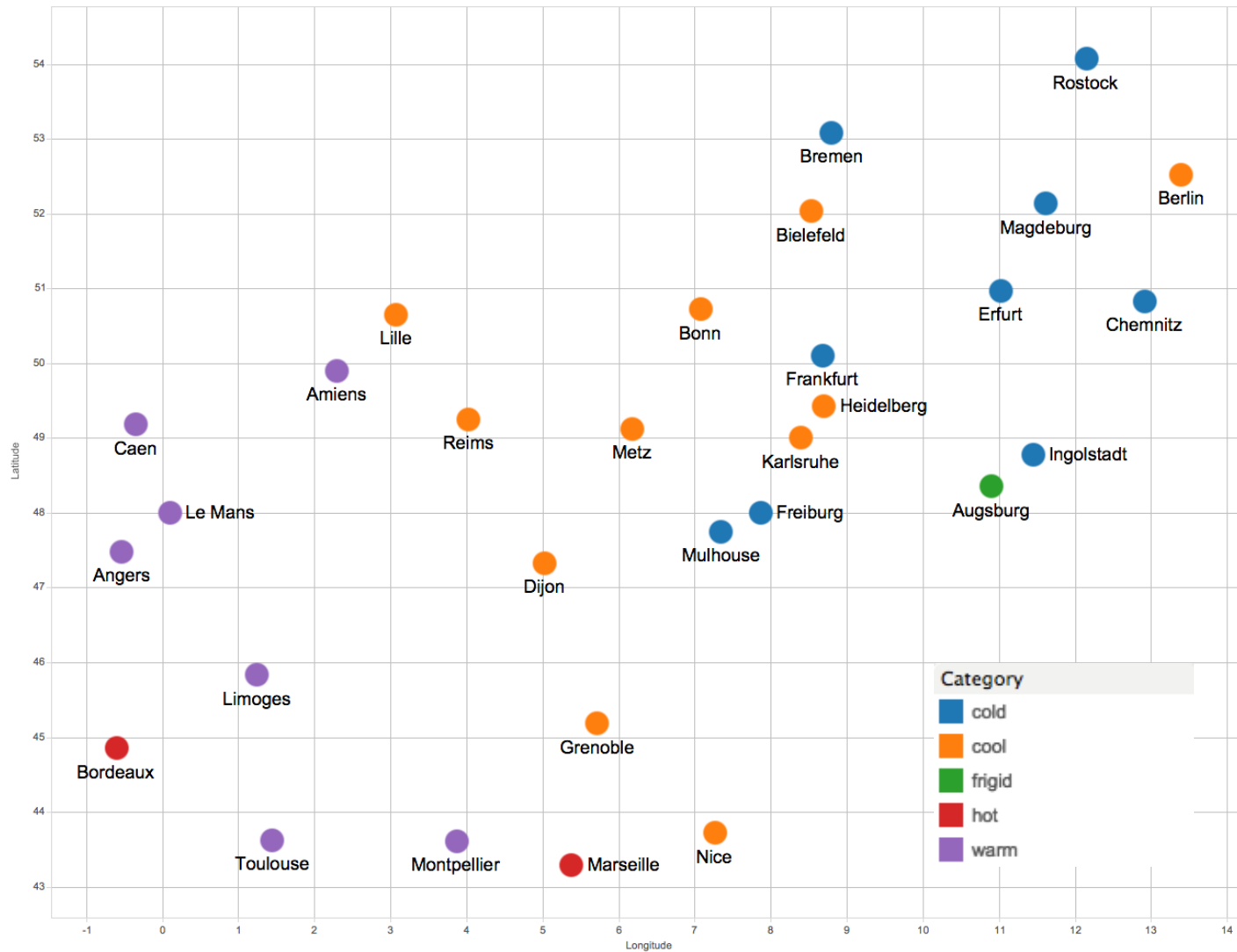
- City temperatures - France and Germany
- Features: longitude, latitude
- Distance is Euclidean distance
$$\text{distance}([o_1, a_1], [o_2, a_2]) = \text{sqrt}((o_1 - o_2)^2 + (a_1 - a_2)^2)$$

= actual distance in x-y plane
- Labels: frigid, cold, cool, warm, hot

Nice (7.27, 43.72) cool  
Toulouse (1.45, 43.62) warm  
Frankfurt (8.68, 50.1) cold  
.....

Predict temperature  
category from  
longitude and latitude

# KNN Example



# KNN Summary

To classify a new item  $i$ : find  $K$  closest items to  $i$  in the labeled data, assign most frequent label

- No hidden complicated math!
- Once distance function is defined, rest is easy
- Though not necessarily efficient
  - Real examples often have thousands of features
    - Medical diagnosis: symptoms (yes/no), test results
    - Email spam detection: words (frequency)

Database of labeled items might be enormous

# “Regression” Using KNN

Features - gender, profession, age, income, postal-code

person<sub>1</sub> = (male, teacher, 47, \$25K, 94305) buyer

person<sub>2</sub> = (female, teacher, 43, \$28K, 94309) non-buyer

Remember training data has labels

To classify a new item  $i$ , find  $K$  closest items to  $i$  in the labeled data, assign most frequent label

person<sub>3</sub> = (female, doctor, 40, \$40K, 95123)

# “Regression” Using KNN

Features - gender, profession, age, income, postal-code

person<sub>1</sub> = (male, teacher, 47, \$25K, 94305) \$250

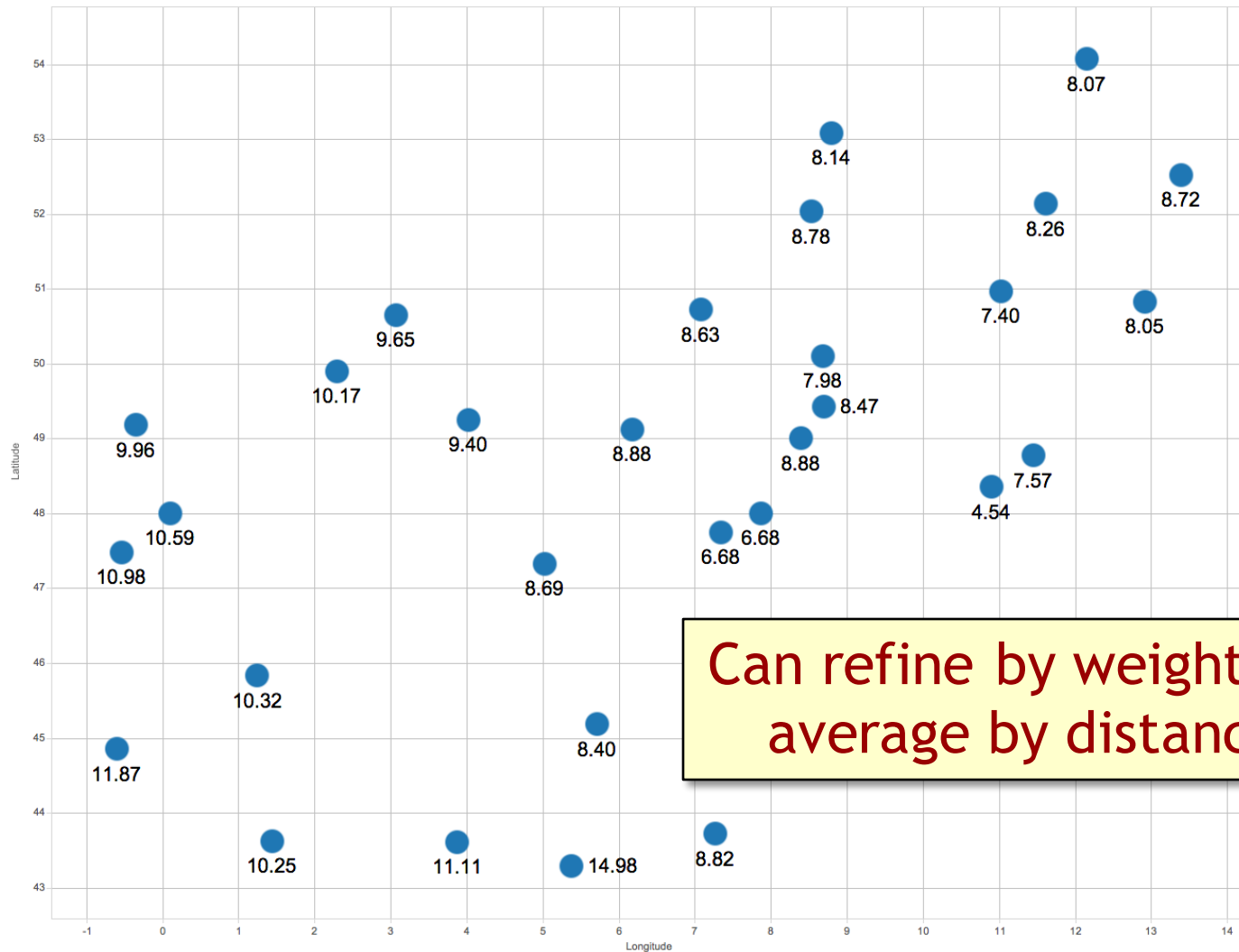
person<sub>2</sub> = (female, teacher, 43, \$28K, 94309) \$100

Remember training data has labels

To classify a new item  $i$ , find  $K$  closest items to  $i$  in the labeled data, assign average value of labels

person<sub>3</sub> = (female, doctor, 40, \$40K, 95123)

# Regression Using KNN - Example



Can refine by weighting  
average by distance



# Decision Trees

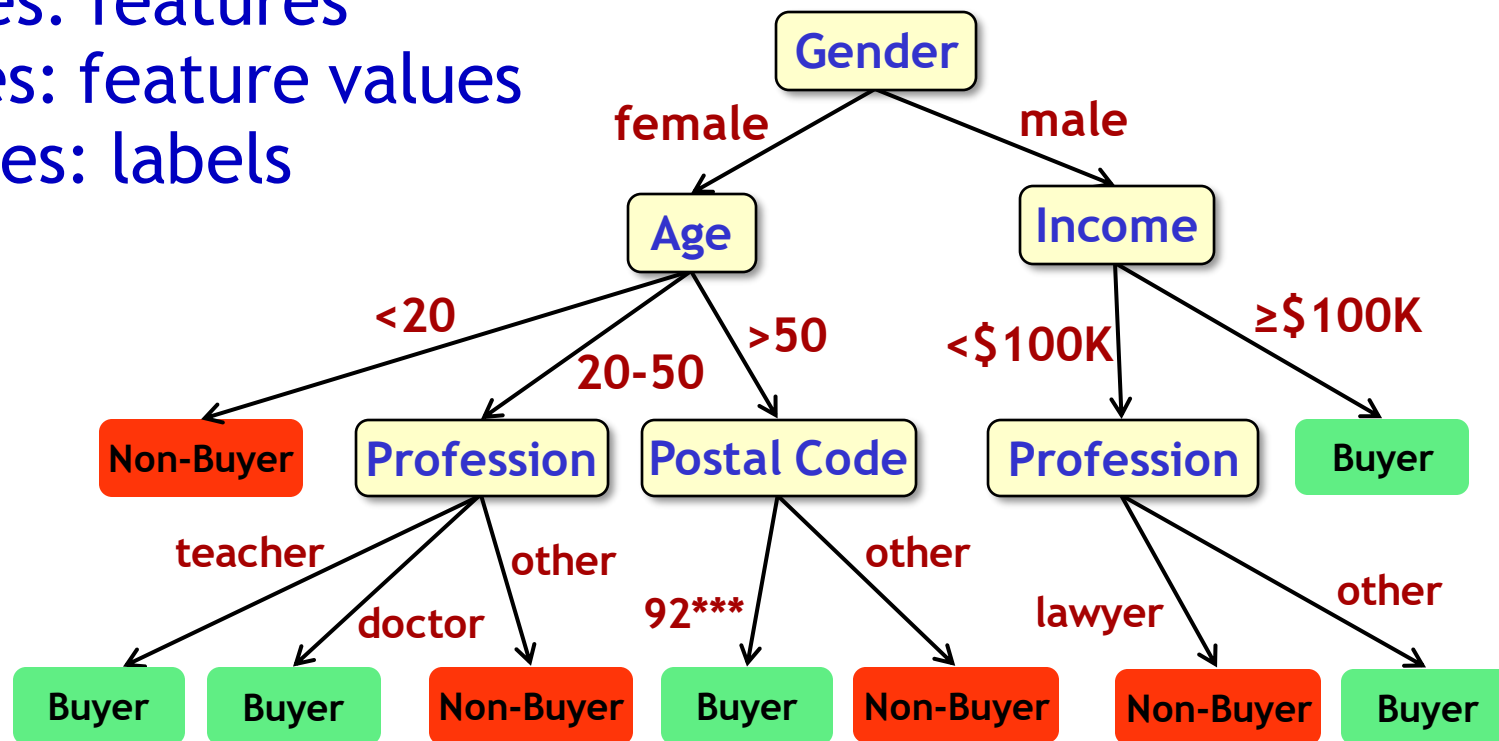
- Use the training data to construct a decision tree
- Use the decision tree to classify new data

# Decision Trees

Nodes: features

Edges: feature values

Leaves: labels



New data item to classify:  
Navigate tree based on feature values

# Decision Trees

Primary challenge is building good decision trees from training data

- Which features and feature values to use at each choice point
- HUGE number of possible trees even with small number of features and values

Common approach: “forest” of many trees, combine the results

- Still impossible to consider all trees

# Naïve Bayes

Given new data item  $i$ , based on  $i$ 's feature values and the training data, compute the probability of each possible label. Pick highest one.

Efficiency relies on conditional independence assumption:

Given any two features  $F_1, F_2$  and a label  $L$ , the probability that  $F_1=v_1$  for an item with label  $L$  is independent of the probability that  $F_2=v_2$  for that item

Examples:

gender and age? income and postal code?

# Naïve Bayes

Given new data item  $i$ , based on  $i$ 's feature values and the training data, compute the probability of each possible label. Pick highest one.

Efficiency relies on **conditional independence assumption**:

Conditional independence assumption often doesn't hold, which is why the approach is “naive”

label  $L$ , the  
h label  $L$  is  
 $F_2=v_2$  for that

item.

Examples:

gender and age? income and

Nevertheless the approach works very well in practice

# Naïve Bayes Example

Predict temperature category for a country based on whether the country has coastline and whether it is in the EU

country	coastline	EU	tempAvg	category
Albania	yes	no	15.18	hot
Andorra	no	no	9.60	warm
Belarus	no	no	5.95	cool
Belgium	yes	yes	9.65	warm
Bosnia and Herzegovina	no	no	9.60	warm
Bulgaria	yes	yes	10.44	warm
Croatia	yes	yes	10.87	warm
Czech Republic	no	yes	7.86	cool
Denmark	yes	yes	7.63	cool
Estonia	yes	yes	4.59	cold
Finland	yes	yes	3.49	cold
Germany	yes	yes	7.87	cool
Greece	yes	yes	16.90	hot
Hungary	no	yes	9.60	warm
Ireland	yes	yes	9.30	warm

# Naïve Bayes Preparation

**Step 1:** Compute fraction (probability) of items in each category

cold	.18
cool	.38
warm	.24
hot	.20



# Naïve Bayes Preparation

**Step 2:** For each category, compute fraction of items in that category for each feature and value

cold (.18)	coastline=yes	.83
	coastline=no	.17
	EU=yes	.67
	EU=no	.33
cool (.38)	coastline=yes	.69
	coastline=no	.31
	EU=yes	.77
	EU=no	.23

warm (.24)	coastline=yes	.5
	coastline=no	.5
	EU=yes	.5
	EU=no	.5
hot (.20)	coastline=yes	1.0
	coastline=no	.0
	EU=yes	.71
	EU=no	.29

# Naïve Bayes Prediction

New item: France, coastline=yes, EU=yes

For each category: probability of category times  
product of probabilities of new item's features  
in that category. Pick highest.

category	prob.	coastline=yes	EU=yes	product
cold	.18	.83	.67	.10
cool	.38	.69	.77	.20
warm	.24	.5	.5	.06
hot	.20	1.0	.71	.14

# Naïve Bayes Prediction

New item: France, coastline=yes, EU=yes

For each category: probability of category times  
product of probabilities of new item's features  
in that category. Pick highest.

category	prob.	coastline=yes	EU=yes	product
cold	.18	.83	.67	.10
cool	.38	.69	.77	.20
warm	.24	.5	.5	.06
hot	.20	1.0	.71	.14

# Naïve Bayes Prediction

New item: Serbia, coastline=no, EU=no

For each category: probability of category times  
product of probabilities of new item's features  
in that category. Pick highest.

category	prob.	coastline=no	EU=no	product
cold	.18	.17	.33	.01
cool	.38	.31	.23	.03
warm	.24	.5	.5	.06
hot	.20	.0	.29	.00

# Naïve Bayes Prediction

New item: Serbia, coastline=no, EU=no

For each category: probability of category times  
product of probabilities of new item's features  
in that category. Pick highest.

category	prob.	coastline=no	EU=no	product
cold	.18	.17	.33	.01
cool	.38	.31	.23	.03
warm	.24	.5	.5	.06
hot	.20	.0	.29	.00

# Naïve Bayes Prediction

New item: Austria, coastline=no, EU=yes

For each category: probability of category times  
product of probabilities of new item's features  
in that category. Pick highest.

category	prob.	coastline=no	EU=yes	product
cold	.18	.17	.67	.02
cool	.38	.31	.77	.09
warm	.24	.5	.5	.06
hot	.20	.0	.71	.0

# Naïve Bayes Prediction

New item: Austria, coastline=no, EU=yes

For each category: probability of category times  
product of probabilities of new item's features  
in that category. Pick highest.

category	prob.	coastline=no	EU=yes	product
cold	.18	.17	.67	.02
cool	.38	.31	.77	.09
warm	.24	.5	.5	.06
hot	.20	.0	.71	.0



# Naïve Bayes Prediction

New item: Austria, coastline=no, EU=yes

For each product in the catalog, we calculate the product of the probabilities of each feature given the product.

Many presentations of Naïve Bayes include an additional normalization step so the final products are probabilities that sum to 1.0. The choice of label is unchanged, so we've omitted that step for simplicity.

category	coastline=no	coastline=yes	EU=yes	EU=no
cool	.38	.31	.77	.09
warm	.24	.5	.5	.06
hot	.20	.0	.71	.0

# Your Turn: World Cup Data

Predict whether team ends in group or knockout stage based on number of yellow cards per game and number of red cards per game

team	games	stage	yellowCards	redCards	yellowPerGame	yellows	redPerGame	reds
Algeria	3	group	4	2	1.33	low	0.67	high
Argentina	5	knockout	7	0	1.40	low	0.00	none
Australia	3	group	7	2	2.33	high	0.67	high
Brazil	5	knockout	7	2	1.40	low	0.40	high
Cameroon	3	group	5	0	1.67	medium	0.00	none
Chile	4	knockout	13	1	3.25	high	0.25	medium
Denmark	3	group	6	0	2.00	medium	0.00	none
England	4	knockout	6	0	1.50	medium	0.00	none
Germany	6	knockout	8	1	1.33	low	0.17	medium
Ghana	5	knockout	11	0	2.20	high	0.00	none
Greece	3	group	5	0	1.67	medium	0.00	none
Honduras	3	group	7	0	2.33	high	0.00	none
Italy	3	group	5	0	1.67	medium	0.00	none
Ivory Coast	3	group	5	0	1.67	medium	0.00	none
Japan	4	knockout	7	0	1.75	medium	0.00	none
Mexico	4	knockout	9	0	2.25	high	0.00	none
Netherlands	6	knockout	15	0	2.50	high	0.00	none

# Your Turn

group (.5)	yellows=low	.20
	yellows=medium	.47
	yellows=high	.33
	reds=none	.60
	reds=medium	.27
	reds=high	.13

knockout (.5)	yellows=low	.33
	yellows=medium	.34
	yellows=high	.33
	reds=none	.67
	reds=medium	.27
	reds=high	.06

1. France: yellows=medium, reds=medium  
group or knockout?
2. USA: yellows=high, reds=none  
group or knockout?

# Feature Management

Real applications often have thousands of features, too many for classification algorithms to handle well

Sometimes useful features are hidden or missing

# Feature Management

Real applications often have thousands of features, too many for classification algorithms to handle well

- **Feature selection** - select subset of features that are independent and predictive
- **Dimensionality reduction** - combine multiple features into one value

*Replace [salary,bonus,options] with income*

*Replace [passes,minutes] with passes-per-minute*

Sometimes useful features are hidden or missing

# Feature Management

Real applications often have thousands of features, too many for classification algorithms to handle well

Sometimes useful features are hidden or missing

- **Feature engineering** - add features from other data or domain knowledge

*distance-from-coast, elevation (for temperature)*

*average player temperament (for yellow and red cards)*

*product ratings from review site*

# Deep Neural Networks

## Neural Networks

- Machine learning method modeled loosely after connected neurons in brain
- Invented decades ago but not successful
- Recent resurgence enabled by:
  - Powerful computing that allows for many layers (making the network “deep”)
  - Massive data for effective training



# Deep Neural Networks

= Deep Learning

- Huge breakthrough in effectiveness and reach of machine learning
- Accurate predictions across many domains
- Big plus: automatically identifies features in unstructured data
  - E.g., images, videos, text
  - “Self-supervised learning”

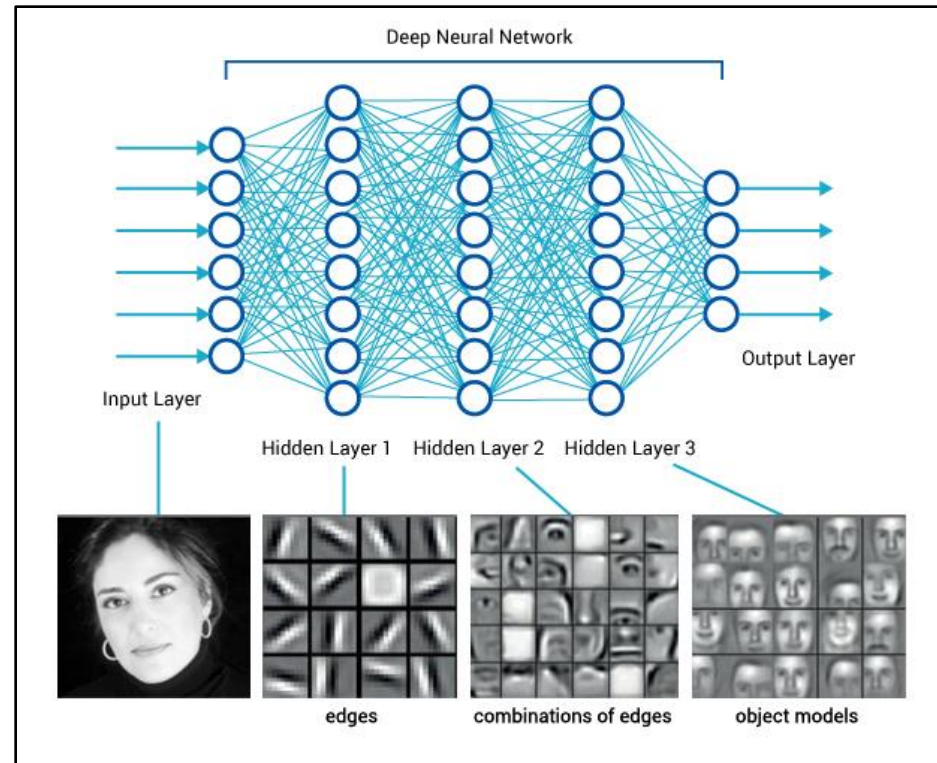
# Deep Neural Networks

## General idea

- Multiple layers, each layer transforms inputs to provide new features or structures for next layer
- Iterate on training data, checking accuracy and improving network

## Reality

- Complex and mysterious, often used without full understanding
- Results not “explainable”



# Large Language Models and Generative AI

- Create Deep Neural Network using massive amount of data
  - Text, images, audio, video, computer code, ...
  - Possibly in a given domain
    - DNN is “large language model” (LLM)
- User issues prompt      New field: “prompt engineering”
- System uses LLM to generate newly assembled content related to the prompt
- Probabilistic: repeating prompt generates new content

# Example: ChatGPT



**You**

Write a two-paragraph advertisement for a data science short-course for students at North South University in Dhaka, Bangladesh taught by Jennifer Widom of Stanford University



**ChatGPT**

🚀 Unlock the World of Data Science Excellence with Prof. Jennifer Widom at North South University, Dhaka! 🚀

Embark on a transformative learning experience with the Data Science Short Course curated exclusively for students at North South University, and taught by the eminent Professor Jennifer Widom from Stanford University. In this dynamic program, you'll delve into the core principles of data science, machine learning, and analytics, gaining practical skills that are highly sought after in today's tech-driven job market. Professor Widom's expertise, honed through years of groundbreaking research, ensures a rich and engaging educational journey, where theory seamlessly integrates with real-world applications.

Join a community of forward-thinkers and equip yourself with the tools to navigate the data landscape confidently. Whether you're a computer science enthusiast, a business major, or an aspiring data scientist, this course offers a unique opportunity to expand your horizons and elevate your career prospects. Don't miss the chance to learn from one of the brightest minds in the field. Enroll now in Jennifer Widom's Data Science Short Course at North South University and position yourself at the forefront of the data revolution! 🌐 📱 #NSUDataScience #JenniferWidomCourse #DataScienceMastery

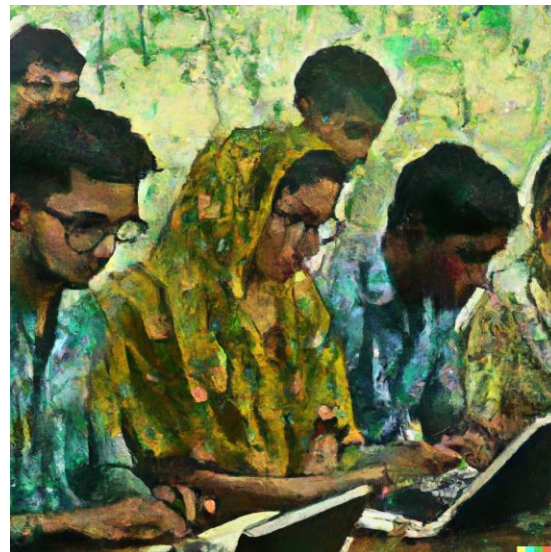




# Example: DALL-E

Bangladeshi students working on data science in the style of van gogh

**Generate**



# Back to Classification: Training and Test

Used training data to create model for making predictions.  
How do you know whether it's a good model?

➤ Try it on known data

Training Data	Feature Values				Labels	Training Data
						"Test Data"

# Confusion Matrix

Full information about results on test data

		Prediction			
		cold	cool	warm	hot
Actual	cold	12	5	2	0
	cool	8	69	12	3
	warm	2	16	57	5
	hot	1	1	9	15

Accuracy  
.718

- Basic accuracy = % correct =  $\Sigma(\text{diagonal}) / \text{total}$
- When numbers or ordinal categories, can also incorporate distance

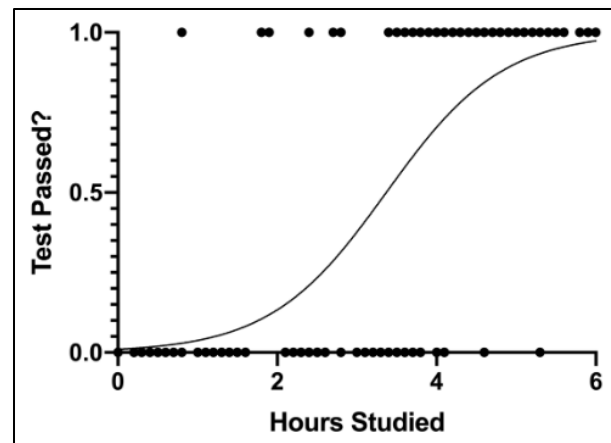


# Other Methods You Might Come Across

## Logistic Regression

- Typically for two labels only (“binary classifier”), call them 0 and 1, and numeric features
- Recall regression model is function  $f$  from input values to numeric output value
- From training data obtain function:

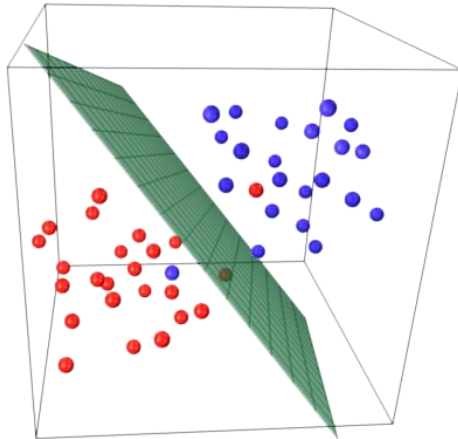
$f(\text{feature-values}) =$   
*probability of output being 1*



# Other Methods You Might Come Across

## Support Vector Machine

- Also for binary classification
- Features = multidimensional space
- From training data SVM finds hyper-plane that best divides space according to labels



# Classification Summary

- Supervised machine learning
- Training data, each example:
  - Set of feature values - numeric or categorical
  - Categorical output value - label
- Model is “function” from feature values to label
  - Use model to predict label for new feature values

# Classification Summary

- Approaches we covered
  - K-nearest neighbors
    - relies on distance (or similarity) function
  - Decision trees
    - relies on finding good trees/forests
  - Naïve Bayes
    - relies on conditional independence assumption
  - Deep neural networks
    - relies on large data sets and powerful computing

# Machine Learning - Classification

Professor Widom's Instructional Odyssey

[www.professorwidom.org](http://www.professorwidom.org)

