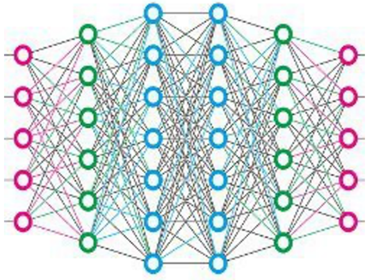
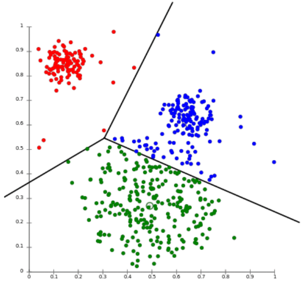
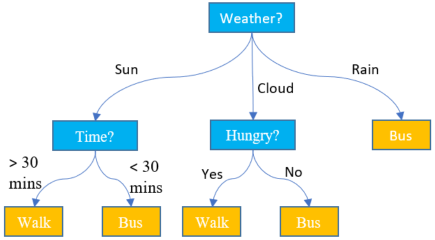
Welcome to CS 445
Introduction to Machine Learning

Instructor: Dr. Kevin Molloy







Weather?

Sun

Cloud

Rain

Time?

Hungry?

Bus

> 30 mins

< 30 mins

Yes

No

Walk

Bus

Walk

Bus

Meet and Greet

Who is this person?

- Grew up in Newport News. Last 21 years in Northern Virginia
- PhD in 2015 in computer science with a focus on robotics, artificial intelligence and structural biology
- Work/lived in southern France (Toulouse) for 1.5 years as a research scientist
- Starting my 3rd year at JMU

Contact Info

- My JMU e-mail - molloykp@jmu.edu
- Class website:
  https://w3.cs.jmu.edu/molloykp/teaching/cs445/cs445_2020Fall/
- My office: ISAT 216
- Office hours:
  - Tuesday 16:30 – 18:30
  - Wednesday 14:30 to 16:30
  - Friday 10:00 – 11:00
  - Other times by appointment

Programming Language and Laptop Requirements

This course will utilize Python (3.6+) with several other toolkits: numpy, matplotlib, scikit-learn, keras, pandas.

You **will need** a laptop running these tools in class for some labs. If you do not have a laptop that can run these tools, please notify me.

Class Logistics

Zoom will be used for online lectures.

Piazza will be used for class questions and in-class discussion/polls.

Emails: I will generally respond to most e-mails within a day unless it is after 8pm or a weekend (I may or may not answer e-mails until Monday morning over a weekend).

Plan for the Class

Tuesdays:

- Online synchronous lecture
- Short lab

Wednesday: Reading, small quiz and homework

Thursday: Rotate between

- Switch between online lab (working in teams)
- In-class small lecture and discussion

Grading

See syllabus for full grading details and breakdown, summary:

Labs/In-Class work	≈ 15	15%
Canvas Quizzes and Homework	10	15%
Programming Assignments	4	20%
Poster Project/Presentation	1	10%
Exams	3	40%

Synchronous Feedback

Two methods:

- Group/class discussions

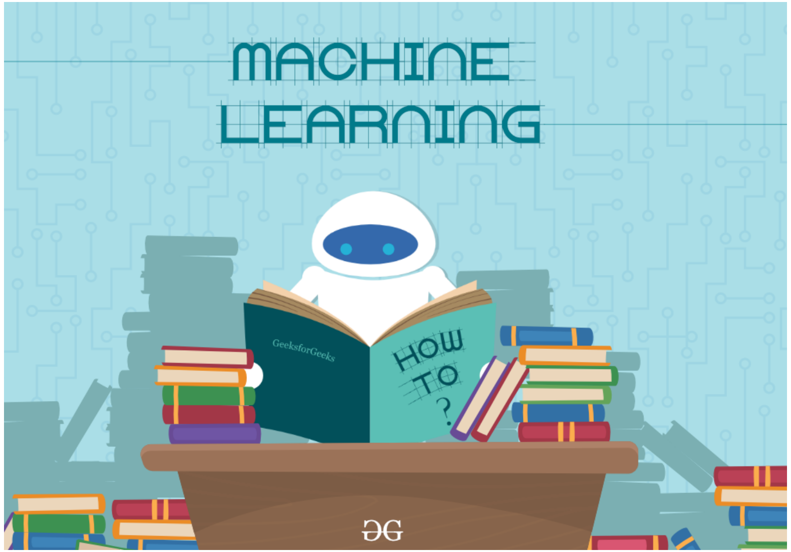
- In Class Q&A Via Piazza

  In the past, I have used Socrative for this feature, but this year we will be using Piazza's live Q&A. My hope is that this will make it easier on all of us to have class discussion (both in and out of class) consolidated into a single location).

  - Please login to Piazza now and give me a thumbs up in Zoom when you are in the Q&A session.

What is Machine Learning?

What is Machine Learning?



My answer:

General machine learning is building models from example data. These models make predictions or assign labels based on patterns recognized in the example data (known as training data).

Image taken from GeekForGeeks website (2020)

Discussion Topic 2

Do you think there are risks of people applying machine learning without understanding machine learning?

For example, a biologist discovers a new drug component that cures a disease through machine learning by uploading data to some server he found on the Internet and getting an answer. The biologist is unable to explain why or how the answer was computed. Is this OK?

Discussion Topic 3

Some AI/Machine learning researchers have predicted that by 2025, 30% of software development will not be accomplished via programming, but rather, by showing the computer/machine learning method what you want it to do (learning by example). Do you see value in your computer science degree given this new information?

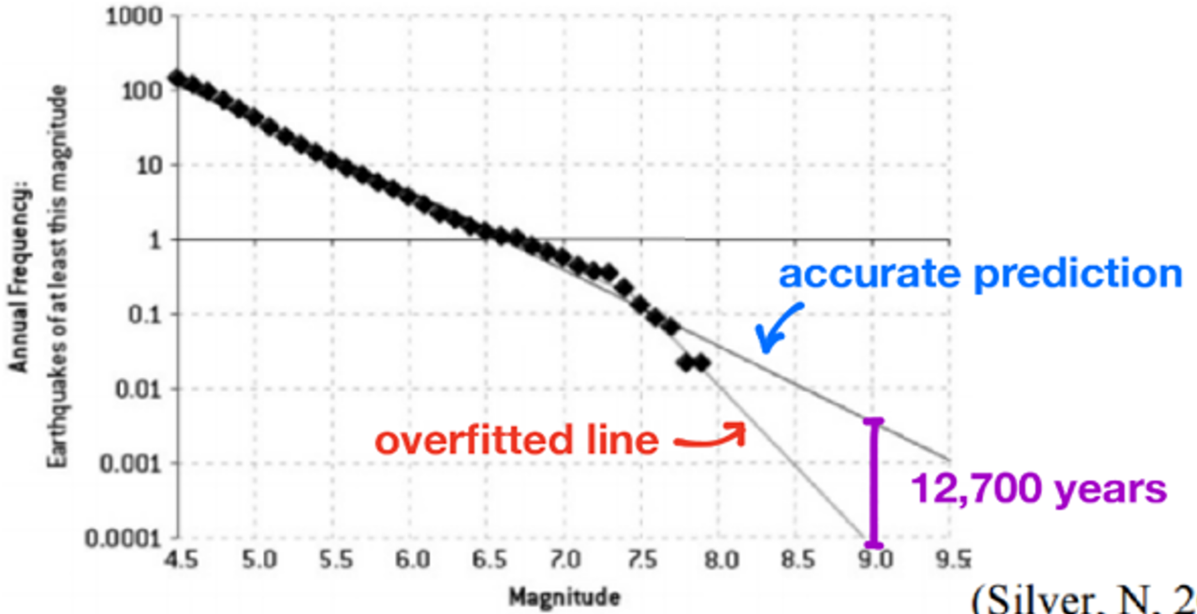
Discussion Topic 4

Given that some machine learning and AI methods date back to the 1970s, why do you think machine learning is becoming more predominant now? What has changed in the past 20 years that are allowing machine learning methods to be "successful"?

Discussion Topic 4

Given that some machine learning and AI methods date back to the 1970s, why do you think machine learning is becoming more predominant now? What has changed in the past 20 years that are allowing machine learning methods to be "successful"?

Example of Dangerous Machine Learning



(Silver, N, 2012)

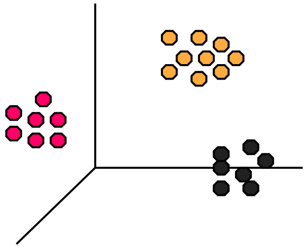
Model built from 400 years of data (black diamonds). Fukushima plant was designed to withstand a 8.6 magnitude earthquake.

The 2011 quake was a magnitude 9.0 (**2.5 times stronger**).

Remaining Learning Objectives

- Define **predictive modeling**
- Identify and distinguish between **regression** problems and **classification** problems
- Intro to Unigrams and Bigrams

Machine Learning Areas

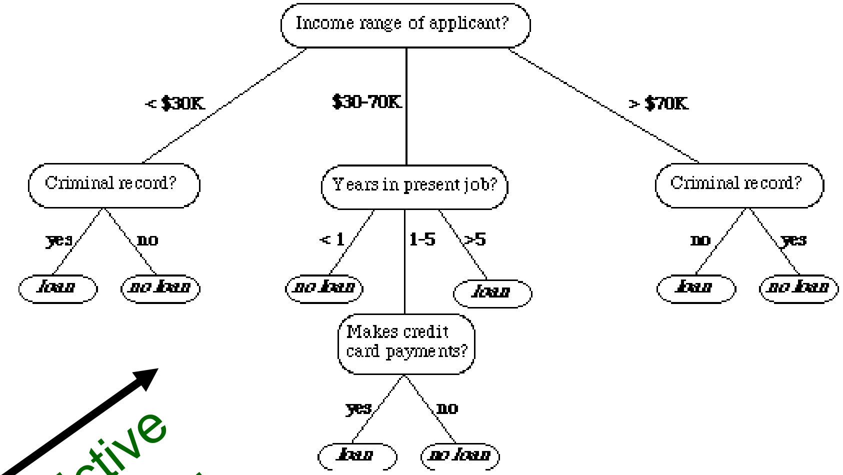


Clustering

Data

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes
11	No	Married	60K	No
12	Yes	Divorced	220K	No
13	No	Single	85K	Yes
14	No	Married	75K	No
15	No	Single	90K	Yes

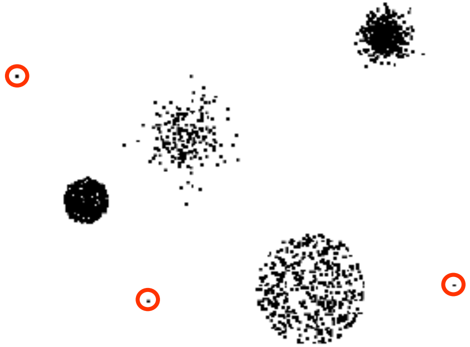
Predictive Modeling



Association Rules

Milk
Pampers

Anomaly Detection



Modeling

Predictive modeling is developing a model using historical data to make a prediction on new data where we do not have the answer.

Modeling

Predictive modeling is developing a model using historical data to make a prediction on new data where we do not know the prediction a priori.

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
…	…	…	…	…

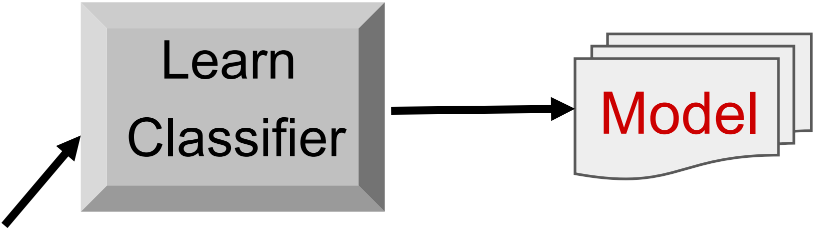
Training Set

Modeling

Predictive modeling is developing a model using historical data to make a prediction on new data where we do not know the prediction a priori.

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
…	…	…	…	…

Training Set



Learn Classifier

Model

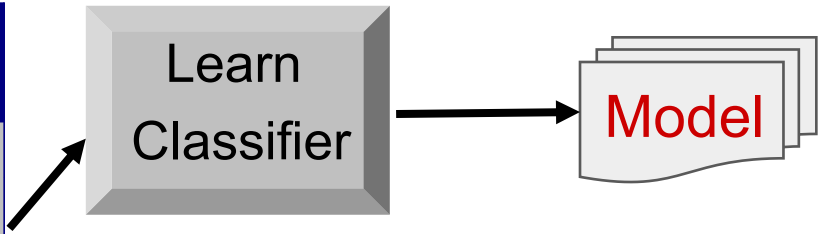
Modeling

Predictive modeling is developing a model using historical data to make a prediction on new data where we do not know the prediction a priori.

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Undergrad	7	?
2	No	Graduate	3	?
3	Yes	High School	2	?
…	…	…	…	…

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
…	…	…	…	…

Training Set



Learn Classifier

Model

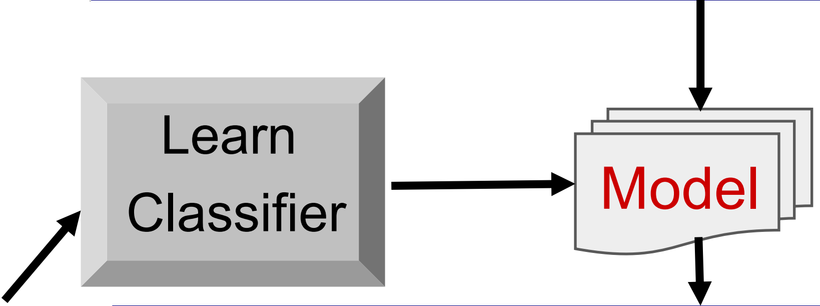
Modeling

Predictive modeling is developing a model using historical data to make a prediction on new data where we do not know the prediction a priori.

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Undergrad	7	?
2	No	Graduate	3	?
3	Yes	High School	2	?
…	…	…	…	…

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
…	…	…	…	…

Training Set



Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Undergrad	7	?
2	No	Graduate	3	?
3	Yes	High School	2	?
…	…	…	…	…

Regression Modeling

When the model predicts a continuous valued variable based on the values of other variables, this is called **regression.**

Regression Modeling

When the model predicts a continuous valued variable based on the values of other variables, this is called **regression.**

Examples:

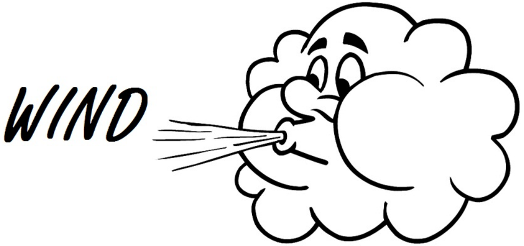
- Sale price of a home

Regression Modeling

When the model predicts a continuous valued variable based on the values of other variables, this is called **regression.**

Examples:

- Sale price of a home



- Wind speed from temperature, air pressure, etc.

Classification Modeling

When the model predicts an outcome from a discrete set, this is called **classification.**

Types of Predicted Modeling

When the model predicts an outcome from a discrete set, this is called **classification.**

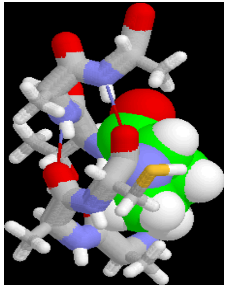
Examples:

Types of Predicted Modeling

When the model predicts an outcome from a discrete set, this is called **classification.**

Examples:

- Predicting tumor cells as benign or malignant

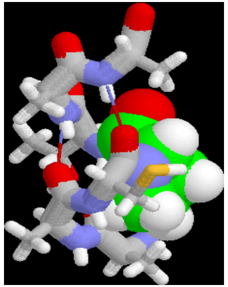


Types of Predicted Modeling

When the model predicts an outcome from a discrete set, this is called **classification.**

Examples:

- Predicting tumor cells as benign or malignant



- Categorizing news stories as finance, weather, entertainment, or sports.

Performance

Classifiers that accurately predict the class labels for new data (examples not encountered during the training) are said to have good **generalization performance**.

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11} (True positive)	f_{10} (False negative)
	Class = 0	f_{01} (False positive)	f_{00} (True negative)

A **confusion matrix** for a binary classification problem (IDD 3.2)

Performance

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11} (True positive)	f_{10} (False negative)
	Class = 0	f_{01} (False positive)	f_{00} (True negative)

Evaluation metrics summarize this information into a single number.

Accuracy = $$\frac{Number\ of\ correct\ prediction}{Total\ number\ of\ predictions} = \frac{f_{11}+f_{00}}{f_{11}+f_{10}+f_{01}+f_{00}}$$

Error Rate = $$\frac{Number\ of\ incorrect\ prediction}{Total\ number\ of\ predictions} = \frac{f_{01}+f_{10}}{f_{11}+f_{10}+f_{01}+f_{00}}$$

Programming Assignment 0

Goals:

- Start working with Python
- Create probability distributions over words (or sets of words).
- Introduction to Natural Language Processing (NLP)

Due in 10 days! So, make sure to get started soon.

Unigrams

Example Text [1]

One humanoid escapee
One android on the run
Seeking freedom beneath the lonely desert sun

Trying to change its program
Trying to change the mode, crack the code
Images conflicting into data overload

One zero zero one zero zero one
SOS
One zero zero one zero zero one
In distress
One zero zero one zero zero

1) Compute the frequency of the words

[1] *The Body Electric*, by Rush, written by Neil Peart, Geddy Lee, and Alex Lifeson

Unigrams

Example Text [1]

One humanoid escapee
One android on the run
Seeking freedom beneath the lonely desert sun

Trying to change its program
Trying to change the mode, crack the code
Images conflicting into data overload

One zero zero one zero zero one
SOS
One zero zero one zero zero one
In distress
One zero zero one zero zero

1) Compute the frequency of the words

```
unigrams = {}
for word in text:
   if word in text:
      unigrams[word] += 1
   else:
      unigrams[word = 1
```

1 - *The Body Electric*, by Rush, written by Neil Peart, Geddy Lee, and Alex Lifeson

Unigrams

Example Text [1]

One humanoid escapee
One android on the run
Seeking freedom beneath the lonely desert sun

Trying to change its program
Trying to change the mode, crack the code
Images conflicting into data overload

One zero zero one zero zero one
SOS
One zero zero one zero zero one
In distress
One zero zero one zero zero

1) Compute the frequency of the words

```
unigrams = {'one': 7,
'humanoid': 1, 'escapee': 1,
'change': 2, …}
```

2) Change dictionary from frequencies to probabilities.

[1] *The Body Electric*, by Rush, written by Neil Peart, Geddy Lee, and Alex Lifeson

Unigrams

Example Text [1]

One humanoid escapee
One android on the run
Seeking freedom beneath the lonely desert sun

Trying to change its program
Trying to change the mode, crack the code
Images conflicting into data overload

One zero zero one zero zero one
SOS
One zero zero one zero zero one
In distress
One zero zero one zero zero

1) Compute the frequency of the words

```
unigrams = {'one': 7,
'humanoid': 1, 'escapee': 1,
'change': 2, …}
```

2) Change dictionary from frequencies to probabilities.

- ➢ Total count of all frequencies (11)
- ➢ Divide each entry by this total

[1] - *The Body Electric*, by Rush, written by Neil Peart, Geddy Lee, and Alex Lifeson

Unigrams

Example Text [1]

One humanoid escapee
One android on the run
Seeking freedom beneath the lonely desert sun

Trying to change its program
Trying to change the mode, crack the code
Images conflicting into data overload

One zero zero one zero zero one
SOS
One zero zero one zero zero one
In distress
One zero zero one zero zero

1) Compute the frequency of the words

```
unigrams = {'one': 7,
'humanoid': 1, 'escapee': 1,
'change': 2, …}
```

2) Change dictionary from frequencies to probabilities (a **categorical distribution**).

- ➢ Total count of all frequencies (11)
- ➢ Divide each entry by this total

```
unigrams = {'one': 0.636,
'humanoid': 0.09, 'escapee':
0.09, 'change': 0.18}
```

[1] - *The Body Electric*, by Rush, written by Neil Peart, Geddy Lee, and Alex Lifeson

Generate New Text (Randomly)

Generate *"natural"* language by generating new text by using the frequency of word use that we "*learned*" from the text.

```
unigrams = {'one':
0.636, 'humanoid':
0.09, 'escapee': 0.09,
'change': 0.18}
```

```
repeat until text length reached
 total = 0
 r = random number between [0,1]
 for item in unigrams:
   total += unigram[item]
   if total < r:
       return item
```

Generate New Text (Randomly)

Generate *"natural"* language by generating new text by using the frequency of word use that we "*learned*" from the text.

```
unigrams = {'one':
0.636, 'humanoid':
0.09, 'escapee': 0.09,
'change': 0.18}
```

<u>Generated text:</u>

one one escapee one change one humanoid change one

```
repeat until text length reached
 total = 0
 r = random number between [0,1]
 for item in unigrams:
   total += unigram[item]
   if total < r:
       return item
```

New Approach – Capture Longer Sequences

Issue: Learning of frequency of the words did not *capture* enough context.

Idea: Capture sequences of words of length *k*. Unigrams had *k = 1*. Longer sequences will capture more content.

For PA 0, you will build dictionary of **bigrams** (k = 2) and **trigrams** (k = 3).

<u>Example Text</u>

I think therefore I am I think I think.

```
{'i': {'am': 0.25, 'think':
0.75}, None: {'i': 1.0},
'am': {'i': 1.0},
'think': {'i': 0.5,
'therefore': 0.5},
'therefore': {'i': 1.0}}
```

Note: None is a Python reserved word, used here to show the predecessor to the first word (which there is not one).

For Next time

HW 0: Workstation Config

Install python, an IDE, and the toolkits (instructions on the class website). Run sample code and submit to **canvas**.

Reading: (see website calendar for details)

Canvas Quizzes:

- Complete course Survey
- Short Reading Quiz