Welcome to CS229 Machine Learning. Uh, some of you know that this class has been taught
at Stanford for a long time.
And this is often the course that, um, I most look forward to teaching each year because this
is where we've helped I think,
several generations of Stanford students become experts in machine learning, go on to build
many of their products and services and startups that I'm
sure many of you or probably all of you are using, uh, today. Um, so what I want to do
today was spend some time talking over, uh, logistics
and then, uh, spend some time, you know, giving you a beginning of a intro, talk a little bit
about machine learning.
So about 229. Um, you know,
all of you have been reading about AI in the news, uh, about machine learning in the news.
Um, and you probably heard me or others say, AI is the new electricity.
Uh, the emergence and rise of electricity about 100 years ago, it transformed every major
industry. Um, I think AI already we call it, machine learning, uh, the rest of the world seems to call AI.
[NOISE] Um, machine learning and deep learning will change the world.
And I hope that through 229, we'll give you the tools you need so that you can be many of
these future titans of industries that you can be one to go out and build, you know, help the
large tech companies do amazing things they do,
or build your own startup, or go into some other industry. Go, go transform healthcare or go
transform transportation or go build a self-driving car.
um, and do all of these things that, um, after this class, I think you'll be able to do.
You know, um, the majority of students supplying- the,
the demand for AI skills- the demand for machine learning skills is so vast. I think you all
know that. Um, and I think it's because machine learning has advanced
so rapidly in the last few years that there are so many opportunities, um, to apply the
learning algorithms, right?
Both in industry as well as in academia. I think today, we have, um, the English department
professors trying to
apply learning algorithms to understand history better. Uh, we have lawyers trying to apply
machine learning to
process legal documents and off-campus, every company, both the tech companies as well
as lots of
other companies that you wouldn't consider tech companies, everything from manufacturing
companies, to healthcare companies,
to logistics companies are also trying, to apply machine learning. So I think that, um, uh, uh,
if you look at it on a- on a factual basis, the number of people doing
very valuable machine learning projects today is much greater than it was six months ago.
six months ago is much greater than it was 12 months ago.
And the amount of value, uh, the amount of exciting meaningful work being done in machine
learning is this, is very strongly going up.
Um, and I think that given the rise of, you know, the,
the amounts of data we have as well as the new machine learning tools that we have, um, it
would be a long time, before society runs out of opportunities.
You know, before we have enough people with the machine learning
20 years ago was a good time to start working,
skillsets. Um, so just as maybe, I don't know, working on this Internet thing and all people that
I think today is a wonderful time to jump into machine learning, uh, and,
started working on the Internet like 20 years ago have fantastic careers.
and the number of- and the opportunities for you to do unique things that no one has- no one
else is doing, right?
The opportunity for you to go to a logistics company and find that exciting way to apply
will be very high, uh,
machine learning, because chances are that logistic company has no one else even working
on this.
Because, you know, they probably can't- they, they may not be able to hire a fantastic
Stanford student that's a graduate of CS229, right?
Because there just aren't a lot of CS229 graduates around. Um, so what I want to do today is,
um, do a quick intro talking about logistics. Um, and then uh, we'll,
we'll spend the second half of the day, you know, giving an overview and, and talk a little bit
more about machine learning.
Okay? And uh, oh, and I apologize. I- I think that, uh, this room, according to that sign there,
seats,
what, 300 and something students. Uh, I think- we have, uh, uh, like not quite 800 people
[LAUGHTER] Uh, so
enrolled in this class. So there are people outside, and, and all of the classes, uh, are recorded
and broadcast in SCPD.
uh, they usually- the videos are usually made available the same day. So for those who they
can't get into the room, my apologies.
Um, there- there were some years, um, where even I had trouble getting into the room but I'm
glad [LAUGHTER] they let me in.

But, but I'm- but, but hopefully, you can watch. You, you can watch all of these things online
shortly. [inaudible]
Oh, I see. Yeah. [LAUGHTER] I don't know, it's a bit complicated. [LAUGHTER] Yeah.
Thank you.
I think it's okay. Yeah. I- I- okay, yeah. Yeah. Maybe for the next few classes you can squeeze
in and use the NTC.
So for now, it might be too complicated. Okay. So quick intros, um,
uh, I'm sorry, I should have introduced myself. My name is Andrew Ng. [LAUGHTER] Uh, uh,
uh, and I wanted to introduce
some of the rest of the teaching team as well. There's a class coordinator.
Um, she has been playing this role for many years now and helps keep the trains run on time
and makes sure that everything in
class happens when it's supposed to. Uh, uh, so, so, so should be uh- and then, uh,
we're thrilled to have- Do you guys want to stand up? Uh, we have the co-head TAs,
uh, respectively are PhD students working with me. Uh, and so bringing a lot of, um, uh,
technical experience, uh, technical experience in machine learning as well as practical
know-how on how to actually make these things work.
And with the large class that we have, we have a large TA team. Um, maybe I won't
introduce all of the TAs here today
but you'll meet many of them throughout this course here. But the TAs expertise span
everything from computer vision to natural language processing,
to computational biology, to robotics. And so, um, through this quarter, as you work on your
class projects,
I hope that you get a lot of, uh, help and advice and mentoring from the TAs, uh, all of which-
all of whom have deep expertise not just in
machine learning but often in a specific vertical application area, um, of machine learning.
So depending on what your projects, we tried to match you to a TA that can give you advice,
um,
the most relevant, uh, whatever project you end up working on. Um, so yeah, goal of this
class,
I hope that after the next 10 weeks, uh, you will be an expert in machine learning.
Um, it turns out that, uh, uh, you know, um,
and- and I hope that after this class, you'll be able to go out and build very meaningful
machine learning applications,
um, either in an academic setting where, uh, hopefully you can apply it to your problems in
mechanical engineering,
electrical engineering, and, uh, English, and law and, um, uh, and- and- and education and
all of this wonderful work that happens on campus,
uh, as well as after you graduate from Stanford to be able to apply it to whatever jobs you
find. Um, one of the things I find very exciting about machine learning is that it's no
longer a sort of pure tech company only kind of thing, right? I think that many years ago,
um, machine learning, it was like a thing that, you know, the computer science department
would do. And that
the elite AI companies like Google and Facebook and Baidu and Microsoft would do. Uh, but
now, it's so pervasive that even companies that are
not traditional tech companies see a huge need to apply these tools, and
find a lot of the most exciting work, uh, these days.
Um, and, um, and maybe some of you guys know my history some would be biased, right?
I led the Google Brain team which helped Google transform
from what was already a great company 10 years ago to today which is, you know, a great AI
company. And then I also led the AI group at Baidu,
and, you know, led the company's technology strategy to help Baidu. Also, it transformed
from what was already a great company
many years ago to today arguably China's greatest AI company. So having led the, you know,
built the teams that led the AI transformations of two large tech companies, I, I, I feel like
that's a great thing to do.
Uh, but even beyond tech, I think that, um, there's a lot of exciting work to do as well to help
other industries,
to help other sectors, uh, embrace machine learning and use these tools effectively. Um, but
after this class,
I hope that each one of you will be well qualified to get a job at a shiny tech company and do
machine learning there,
or go into one of these other industries and do very valuable machine learning projects
there. Um, and, um, and also,
many of you, um, are taking this class with the primary goal of, uh,
being able to do research, uh, in machine learning, so, so, actua- so some of you I know are
PhD students.
Um, I hope that this class will also leave you well-equipped to, um, be able to read and
understand research papers, uh,
as well as, uh, you know, be qualified to start pushing forward, um, the state of the art, right.
Um, so let's see. Um, so today, uh,

so, so just as machine learning is evolving rapidly, um, the whole teaching team,
we've been constantly updating CS229 as well. So, um, it's actually very interesting.
I feel like the pace of progress in machine learning has accelerated, so it, it actually feels like
that, the amounts we changed the class year over year has been increasing over time.
um,
So- so for your friends who took the class last year, you know, things are a little bit different
this year because we feel like the whole field of machine
to keep up with what feels like substantially accelerating progress in
learning. Um, so, so, we're no more- um, what we used to hand out paper copies of
For example, uh, so, we've done some, uh, logistical changes.
handouts, we're, we're trying to make this class digital only. Uh, but let me talk a little bit about,
uh, that's,
uh, prerequisites as well as in case your friends have taken this class before,
some of the differences for this year, right? Um, so prerequisites.
Um, we are going to assume that, um, all of you have a knowledge of basic computer skills
and principles.
uh, so, you know, Big O notation, queues, stacks, binary trees. Hopefully, you understand
what all of those concepts are.
And, uh, assume that all of you have a basic familiarity with, um, uh, probability, right?
Hopefully, you know what's a random variable, what's the expected value of a random
variable, what's the variance of a random variable.
Um, and, if- for some of you, maybe especially the SCPD students taking this remotely,
it has been, you know, some number of years since you last had a probability and statistics
class, uh, we will have, uh, review sessions, uh,
Um, on Friday, where we'll go over, uh, some of this prerequisite material as well. But it's
okay. Hopefully, you know. But if you're a little bit fuzzy on those concepts, we'll go over them
what expected value is. what a random variable is,
Also, seem to be a familiar with basic linear algebra. So hopefully that you know what's a
again, um, at a discussion section, uh, on Friday.
matrix, what's a vector, how to multiply two matrices and multiply a matrices and a vector.
but, if you know what an eigenvector is then that's even better. Uh, if you're not quite sure what
an eigenvector is,
we'll go over it that- that you, you, you better- uh, yeah, we'll, we'll, we'll go over it I guess.
And then, um, a large part of this class, uh, uh, is,
um, having you practice these ideas through the homeworks, uh, as well as I mention later a,
And open-ended project.
Uh, so, um, deep, there are- we've actually, uh, until now we used to use uh MATLAB,
uh and Octave for the programming assignments, uh, but this year, we're trying to shift the
programming assignments to, uh,
Python, um, and so, um, I think, uh, for a long time, uh, even today, you know,
I sometimes use Octave to prototype because the syntax in Octave is so nice and just run,
you know, very simple experiments very quickly. But I think the machine learning world, um,
is,
you know, really migrating I think from a MATLAB Python world to increasing- excuse me,
MATLAB Octave world to increasingly a Python maybe and- and then eventually for
production, Java or C++, kind of world.
And so, uh, we're rewriting a lot of the assignments
for this class this school year. Having, having driving that process, uh, so that- so that this
Uh, so, maybe- um, could maybe all of the assignments.
course, um, you could do more of the assignments in, um, Python, uh, NumPy instead.
Um, now, a note on the honor codes, um, we ask that, you know,
we, we actually encourage you to form study groups. Uh, so, so you know I've been um,
spent a long time studying education.
fascinated by education and pedagogy and how instructors like us can help
And one of the lessons I've learned from the educational research literature is that the highly
support you to learn more efficiently.
technical classes like this,
if you form study groups, uh, you will probably have an easier time, right? So, so CS229, we
go for the highly technical material.
There's a lot of math, some of the problems are hard and if you have a group of friends to
study with, uh,
you probably have an easier time uh, uh, because you can now ask each other questions
and work together and help each other.
Um, where we ask you to draw the line or what we ask you to, to, to do relative to Stanford's,
uh, Honor Code is, um, we ask that you do the homework problems by yourself, right?
Uh, and, and, and most specifically, um, it's okay to discuss the homework problems with
friends, but if you, um,

but after discussing homework problems with friends, we ask you to go back and write out
the solutions by yourself, uh,
without referring to notes that, you know, you and your friends had developed together,
okay? Um, the class's honor code is written clearly
on the class handouts posted digitally on the website. So if you ever have any questions
about what
is allowed collaboration and what isn't allowed, uh, please refer to that written document on
the course website where we describe this more clearly, but,
um, out of respect for the Stanford honor code as well as for, uh, uh, you know,
for, for, for students kind of doing their own work, we asked you to basically do your own
work, uh, for this- it's okay to discuss it,
but after discussing homework problems with friends, ultimately, we ask you to write up your
problems by yourself so that
the homework submissions reflect your own work, right? Um, and I care about this because
it turns out that having CS 229,
you know, CS 229 is one of those classes that employers recognize. Uh- uh, I don't know if
you guys know,
but there have been, um, companies that have put up job ads that say stuff like,
"So long as you've got- so long as you completed CS 229 we guarantee you get an
interview, right?
[LAUGHTER] I've- I've seen stuff like that. And so I think you know in order to, to maintain
that sanctity of what it means to be a CS 229 completer,
I think, um, and I'll ask all of you to really do your own homework. Um, or stay within the
bounds of acceptable,
acceptable collaboration relative to the honor code. Um, let's see.
And I think that um, uh, if- uh, you know what?
This is, um, [NOISE] yeah. And I think that, uh, one of the best parts of CS 229,
it turns out is, um, excuse me. So I'm trying, sorry, I'm going to try looking for my mouse
cursor.
Uh, all right. Sorry about that.
My- my- my displays are not mirrored. So this is a little bit awkward.
Um, so one of the best parts of the class is- oh,
shoot. Sorry about that. [LAUGHTER] All right, never mind. I won't do this.
Um, you could do, you could do it yourself online later. Um, yeah, I started
using- I started using Firefox recently in addition to Chrome here. It's a mix up.
Um, one of the best parts of, um, the class is, um, the class project.
Um, and so, you know, one of the goals of the class is to leave you well-qualified to do a
meaningful machine learning project.
And so, uh, one of the best ways to make sure you have that skill set is through this class
and hopefully with the help of some of our TAs.
uh, we really support you to work in a small group to complete a meaningful machine
learning project.
Um, and so one thing I hope you start doing, you know, later today, uh, is to start
brainstorming maybe with your friends.
Um, some of the- some of the class projects you might work on. Uh, and the most common
class project that, you know,
people do in CS 229 is to pick an area or pick an application that excites you and
to apply machine learning to it and see if you can build a good machine learning system for
some application.
And so, uh, if you go to the course website, you know, cs229.stanford.edu and look at
previous years' projects,
you, uh- you, you see machine learning projects applied to pretty much, you know, pretty
much every imaginable application under the sun.
Everything from, uh, diagnosing cancer to creating art to, uh, lots of, um, uh,
applying it to application areas, um, uh,
projects applied to other areas in EE, or Mechanical engineering, or Civil engineering, or
to applying it to understand literature,
Earthquake engineering, and so on, uh, to applying it to um, uh, I don't know.
And, and, and, and, and so, uh, if you look at the previous year's projects of many of which
are posted on the course website.
You could use that as inspiration to see the types of projects students complete, completing
this class are able to do and also encourage you to, um, uh,
you can look at that for inspiration to get a sense of what you'll be able to do at the end-
conclusion of this class and also see if looking at
previous year's projects gives you inspiration for what, um, you might do yourself.
Uh, so we asked you to- we, we invite you I guess to do class projects in small groups and
so, um,
after class today, also encourage you to start making friends in the class both for the
purpose of forming

study groups as well as with the purpose of maybe finding a small group to do a class project with.
Um, uh, we asked you to form project groups of, um, up to size three.
Uh, uh, most project groups end up being size two or three. Um, if you insist on doing it by yourself,
right without any partners that's actually okay too. You're welcome to do that. But, uh, but- but I think often, you know,
having one or two others to work with may give you an easier time. And for projects of exceptional scope,
if you have a very, very large project, that just cannot be done by three people. Um, uh, let us know, and we're
sometimes, you know, we've open to- with, with to some project groups of size four,
but our expectation- but we do hold projects, you know, with a group of four to a higher standard than projects with size one to three, okay.
So- so what that means is that if your project team size is one, two or three persons, the grading criteria.
If your project group is bigger than three persons, we use a stricter criteria when it comes to grading class projects.
Okay. Um, and that, that reminds me um, uh, I know that uh- let's see.
So for most of you since this- since this started 9:30 AM on the first day of the quarter, uh, for many of you,
this may be- this may be your very first class at Stanford. How many of you, this is your very first class at Stanford?
Wow. Cool. Okay. Awesome. Great. Welcome to Stanford. [LAUGHTER] Uh, and if someone next to
you just raise their hand- uh actually, rai- raise your hand again. So I hope that, you know, maybe after class today, if someone next to you raised their hand, you
help welcome them to Stanford, and then, say hi and introduce yourself and make friends on the way.
Um. [NOISE] Yeah. Cool. Nice, nice to see so many of you here.
All right. So um, just a bit more on logistics, uh-
So, um, let's see, in addition to the main lectures that we'll have here,
uh, on Mondays and Wednesdays, um, CS229 also has discussion sections, uh,
on- held on Fridays that are- and everything we do including the- all the, all the lectures and discussion sections are recorded and broadcast through SCPD,
uh, through the online websites. Um, and, uh, discussion sections are taught,
uh, usually by the TAs on Fridays and attendance at discussion sections is optional.
Uh, and what I mean is that, um, you- you know, you- 100% promise, there won't be material on the midterm that will sneak in from the discussion section.
So it's 100% optional. Uh, and you will be able to do all the homework and the projects without attending the discussion section.
But what we'll use the discussion section for, uh, for the first three discussion sections. So, you know, this week, next week, uh, the week after that,
we'll use the discussion sections to go over prerequisite material in greater depth. So, uh, go over, uh,
linear algebra, basic probability statistics, teach a little bit about Python NumPy in case you're less familiar with those frameworks.
Uh, so we'll do that for the first few weeks. And then for the discussion sections that are held later this quarter,
we'll usually use them to go over more advanced optional material. Uh, for example, um,
CS229, most of the learning algorithms, you- you hear about in a class rely on convex optimization algorithms,
but we want to focus the class on the learning algorithms and spend less time on convex optimization.
So if you want to come and hear about more advanced concepts in convex optimization. We'll defer that to the discussion section.
Uh, and then, there are few other advanced topics, uh, Hidden Markov Models, time series,
uh, that we're planning to defer to the, um, Friday discussion sections.
Okay. Um, so, uh, let's see.
Um, cool, and, uh, and, um, a final bit of logistics, um, uh,
for- there are digital tools that some of you have seen, but, um, for this class, we'll drive a lot of the discussion through the online website Piazza.
Uh, how- how many of you have used Piazza before? Okay, cool, That's very amazing.
Mostly. Wow, alright. Uh, good. So, so, uh, online discussion board
for those of you that haven't seen it before, but, um, I definitely encourage you to participate actively
on Piazza and also to answer other student's questions. I think that one of the best ways to
learn as well as contribute, you know,

back to the class as a whole is if you see someone else ask a question on Piazza, if you
jump in and help answer that, uh,
that, that often helps you and helps your classmates. I strongly encourage you to do that.
For those of you that have a private question, you know, sometimes we have students, um,
uh,
reaching out to us to- with a personal matter or something that, you know,
is not appropriate to share in a public forum in which case you're welcome to email us at the
class email address as well.
uh, and we answers in, in the class email address- the cla- teaching staff's email address on
the course website, you can find it there and contact us.
But for anything technical, anything reasonable to share with the class, uh, which includes
most technical questions and most logistical questions, right?
Questions like, you know, can you confirm what date is midterm, or, or, you know, what
happens? Uh, can you confirm who is the team for this going on and so on?
For questions that are not personal or private in nature, I strongly encourage you to post on
Piazza rather than emailing us because, statistically,
you actually get a faster answer, uh, post- posting on Piazza than- than, you
know,
if you, wait for one of us to respond to you, um, and we'll be using Gradescope as well, um,
to- for, for, online grading.
And then, if, if you don't know what Gradescope is, don't worry about it. We'll, we'll, we'll
send you links and show you how to use it later.
Um, oh, and, uh, again, relative to- one last logistical thing to plan for, um,
unlike previous, um, uh, years where we taught CS229,
uh, so we're constantly updating the syllabus, right? The technical content to try to show you
the latest machine learning algorithms, um,
and the two big logistical changes we're making this year, I guess one is, uh,
Python instead of MATLAB, and the other one is, um, instead of having a midterm exam, you
know,
there's a timed midterm, uh, we're planning to have a take-home midterm, uh, this course,
instead.
So, I know some people just breathed in sharply when I said that. [LAUGHTER] I don't
know what that means.
[LAUGHTER] Was that shock or happiness? I don't know. Okay. Don't worry, midterms are
fun. You- you'll, you'll love it.
[LAUGHTER] All right. So that's it for the- that's it for the logistical aspects.
Um, let me check with the- so let- let me check if there are any questions.
Oh, yeah, go ahead. On campus, are those courses offered every quarter [inaudible].
Yeah. So that's interesting. Uh, let's see. I think it's offered in spring.
And one other person. Oh, yes, is teaching it. So someone else is teaching it in spring
quarter.
Um, uh, I actually did not know it was gonna be offered in winter.
[inaudible] Yeah. [inaudible]. Yeah, right, yeah. So- so I think a free guide and teaching it in-
sorry,
in their [inaudible] and you are right, are teaching it in, uh, spring, uh, and I don't think it is
[inaudible]
offered in winter.
Will the session be recorded? Yes, they will be. Oh, and by the way, if, if, if you wonder why
I know it feels weird, I'm repeating the question,
I'm recording the- with the microphone, so that- so that people watching this at
home can hear the question. But, uh, both the lectures and the discussion sections,
uh, will be- will be recorded and put on the website. Uh, maybe the one thing we do, that's not
recorded and broadcast are the office hours. Great. Isn't that right? [LAUGHTER] Oh, oh, but,
uh, I think, uh, this year, uh, we have a 60-hour, how many hour? Well, 60 office hours.
Uh, 60 office hours per week. Right, yeah. [LAUGHTER] So- so- so hopefully, I- I just again,
we- we're constantly trying to improve the course. In previous years, one of the feedback we
got was that the office hours were really crowded.
So- so we have 60 hour- 60 hours, about 60 office hour slots per week this year. That- that
seems like a lot.
So hopefully, if you need to track down one of us, track down a TA to get help, hopefully, that-
[inaudible]. Say that again. Well-
that'll make it easier for you to do so. Go ahead.
[inaudible]. Oh, well logistical things like when homeworks are due, would be covered in
lectures.
Uh, we have uh, yes, so we have uh, four planned homeworks. Oh sorry. [inaudible]
Yeah, and if you go to the- if you go to the course website and you click on the syllabus link
uh, that, has a calendar with when each homework assignments go out and when they'll be
due.

Uh, so four homeworks and uh, project proposals due a few weeks from now and uh,
final projects due at the end of the quarter. But all the, all the exact days are listed on the
course website, I think.
[inaudible] Uh, sure yes, difference between this class and 229a.
Um, let me think how to answer that. Yes. Uh, so yeah I know,
I was debating earlier this morning how to answer that because I've been asked that a few
times. Um, so I think that what has happened at Stanford is that,
the volume of demand for machine learning education is just, right skyrocketing because
everyone wants to learn,
anything everyone sees this stuff and so um, uh, so within- so
the computer science department has been trying to grow the number of machine learning
offerings we have. Um, uh, we actually kept the enrollment to
CS229a at a relatively low number at 100 students. So I actually don't want to encourage too
many of you to sign up because um,
I think we might be hitting the enrollment cap already so, so please don't all sign up for
CS229a because um,
we- CS229a does not have the capacity this quarter but since CS229a is uh, um,
much less mathematical and much more applied, uh, uh, a relatively more applied version of
machine learning,
so, I guess I'm teaching and uh, CS229a and CS230 and CS229, this quarter. Of the three, CS229,
is the most mathematical. Um, it is a little bit less applied than CS229a which is more applied
Machine learning and CS230 which is deep learning.
My advice to students is that um, CS229, uh, CS229a, excuse me, let me write this down.
I think I'm- so CS229a, uh,
is taught in a flipped classroom format which means that, uh, since taking it, we'll mainly
watch videos um,
on the Coursera website and do a lot of uh, programming exercises and then, meet for
weekly discussion sections.
Um, but there's a smaller class with [inaudible] . Um, I, I would advise you that um,
if you feel ready for CS229 and CS230 to do those uh, but CS229,
you know, because of the math we do, this is a, this is a very heavy workload and pretty
challenging class and so,
if you're not sure you're ready for CS229 and CS229a, it may be a good thing to,
to, to take first, uh, and then uh, CS229, CS229a
cover a broader range of machine learning algorithms uh, and CS230 is more focused on
deep learning algorithms specifically, right.
Which is a much narrow set of algorithms but it is, you know, one of the hardest areas of
deep learning.
Uh, there is not that much overlap in content between the three classes. So if you actually
take all three,
you'll learn relatively different things from all of them uh, in the past, we've had students
simultaneously, take 229 and 229a and there is a little bit of overlap.
You know, they, they kind of cover related algorithms but from different points of view. So,
so, some people actually take multiple of those courses at the same time.
Uh, but 229a is more applied, a bit more, you know, practical know-how hands-on and so on
and,uh, much less mathematical. Uh, and, and CS230 is also less
mathematical more applied more about kind of getting it to work where CS229a, um, we do
much more mathematical derivations in CS229.
Cool, any questions? Yes, someone had their hand up.
[inaudible]
So uh, once you say that what- I would generally prefer students not do that in the interest of
time but what, what do you want?
[inaudible] Oh, I see, sure go for it.
Who is enrolled in 229 and 230? Oh not that many of you, interesting.
Oh, that's actually interesting. Cool. Yeah. Thank you, yeah, I just didn't want to set the
presence of students using this as a forum to run surveys.
[LAUGHTER] That was, that was, that was an interesting question. So thank
you. [LAUGHTER]
Um, cool. All right, and, and by the way I think uh, you know,
just one thing about Stanford is the AI world and machine learning world, AI is bigger than
machine learning right
and machine learning is bigger than deep learning. Um, one of the great things about being
a Stanford student is,
you can and I think should take multiple classes, right. I think that your CS229,

has for many years been the core of the machine learning world at Stanford. Uh, but even
beyond CS229,
it's worth your while to take multiple classes and getting multiple perspectives. So, so if you
want to uh,
be really effective, you know, after you graduate from Stanford, you do wanna be an expert in
uh, and you probably wanna know probability statistics. Maybe.
machine learning. You do wanna be an expert in deep learning, you wanna know a bit of
convex optimization, and maybe you
want to know a bit more about reinforcement learning, know a ittle bit about planning, know a
bit about lots of things.
So, I actually encourage you to take multiple classes I guess.
Cool. All right. Good. Um, if there are no more questions,
let's go on to talk a bit about some machine learning.
So um, all right, so the remainder of this class,
what I'd like to do is um, give a quick overview of uh,
you know, the major uh, areas of machine learning and also um,
and, and also give you a sort of overview of the things you learn uh, in the next 10 weeks.
So, you know, what is machine learning? Right. It seems to be everywhere these days and
it's useful for so many places,
and, and I think that um, and uh, you know, and uh, uh,
and I, I feel like they uh- just to share with you my personal bias, right. You, you read the
news about these people making
so much money building learning algorithms. I think that's great. I hope, I hope all of you go
make a lot of money,
but the thing I find even more exciting is, is the meaningful work we could do. I think that, you
know,
I think that every time there's a major technological disruption which there is now, through
machine learning um,
it gives us an opportunity to remake large parts of the world and if we behave ethically in a
you know, help people use these superpowers of machine learning, uh, to do things you can,
principled way, help people live, right. Maybe we could um, uh, maybe we can improve the
healthcare system
maybe you can improve give every child a personalized tutor. Uh, maybe we can make our
democracy run better rather than make it run worse.
But I think that um, the meaning I find in machine learning is that there's so many people that
are so eager for us to go in and help them with these tools that um, if, if you become good at
these tools,
it gives you, an opportunity to really remake some piece, some meaningful piece of the world.
Uh, hopefully in a way that helps other people and makes the world kind of, makes the world
a better place, you know, with these tools,
But, but I think, so, in Silicon Valley, you actually have the power to do that and if you
go make a lot of money, that's great too.
But I find um, much greater meaning in the work we could do. Um, it gives us a unique
opportunity to do all these things, right.
But um, despite the excitement of machine learning. What is machine learning? So let me
give you a couple um,
definitions of machine learning. Um, Arthur Samuel whose claim to fame was uh,
building a checkers playing program, uh, defined it as follows. So field of study gives
computers the ability to learn without being explicitly programmed.
Um, and so you know, interestingly, when Arthur Samuel many,
many decades ago, built the checkers playing program. Uh, the debates of the day was can
a computer ever do something
that it wasn't explicitly told to do? And Arthur Samuel uh, wrote a
checkers playing program, that through self play learns whether the patterns of uh,
the checkerboard that are more likely to lead to win versus more likely to lead to a loss and
learned uh,
to be even better than Arthur Samuel the author himself at playing checkers.
So back then, this was viewed as a remarkable result that a computer programmer, you
know, that could write a piece of
software to do something that the computer program himself could not do, right, because
this program became better than Arthur Samuel.
at, uh, um, at, at the task of playing checkers. um,
Um, and I think today we um, are used to computers or machine learning algorithms
outperforming humans on so many tasks.
Uh, but it turns out that when you choose a narrow task like, speech recognition on a certain
type of task,
you can maybe surpass human level performance. If you choose a narrow task like, playing
the game of Go,

than by throwing really, tons of computational power at it and self play. Uh, uh, uh you can
have a computer,
you know, become very good at, at these narrow tasks. But this is maybe one of the first such
examples in the history of computing. Um.
Uh, and I think this is one of the most widely cited, um, definitions right.
Gives computers the ability learn without being explicitly programmed. Um, my friend Tom
Mitchell in his textbook,
defined this as a Well-posed Learning Problem. Uh, a program is said to learn from
experience E with respect to task T and some performance measure P, if its performance on
T, as measured by P, improves with experience E. And I- I asked Tom this.
I asked Tom if, um, he wrote this definition just because he wanted it to rhyme and
[LAUGHTER] he, he, um, but he did not say yes,
I don't know, um, in this definition, the experienced E. For- for the case of playing
checkers,
the experience E would be the experience of having a checkers play- program played tons of
games against itself.
Uh, so computers lots of patients and sit there for days playing games or checkers against
itself.
So that's experience E. The task T is the task of playing checkers, the performance measure
P, maybe,
Um, what's the chance of this program winning the next game of checkers it plays against the
next opponent.
Right. So- so we say that, ah, this is a well-posed learning problem, learning the game of
checkers.
Now, within this, um, set of ideas with machine learning, there are many different tools we
use in machine learning.
And so in the next 10 weeks, you'll learn about a variety of these different tools.
Um, and so the first of them and the most widely used one is supervised learning.
Um, let's see. I wanna switch to the white board. Do you guys know how to raise the screen?
[NOISE]
So what I wanna do today is really go over some of the major categories of, uh, Machine
Learning Tools, and, um,
and so what you learn in the next, um, ah, by the end of this quarter.
So the most widely used machine learning tool is,
uh, today is supervised learning. Actually, let me check, how, how many of you know what
supervised learning is? Ah, like two-thirds, half of you maybe.
Okay cool. Let me just briefly define it. Um, here's one example. Let's say, you have a
database of
housing prices and so I'm gonna plot your dataset where on the horizontal axis, I'm- I'm
gonna plot the size of the house in square feet.
And on the vertical axis, we'll plot the price of the house. Right. And, um, maybe your dataset
looks like that.
Um, and so the horizontal axis, I guess we'd call this X and vertical axis we'll call that Y.
So, um, the supervised learning problem is given a dataset like this to find the relationship
mapping from X to Y.
And so, um, for example, let's say- let's say- let's say you have- let's say you're fortunate
enough to own a house in Palo Alto,
Right. Ah, and you're trying to sell it, and you want to know how the price of the house. So
maybe your house has a size,
you know, of that amount on the horizontal axis. I don't know, maybe this is 500 square feet,
1,000 square feet, 1,500 square feet. So your house is, ah, 1,250 square feet.
Right. And you want to know, you know, how do you price this house. So given this dataset,
one thing you can do is, um, fit a straight line to it. Right. And then you could estimate or
predict the price
to be what the value you read off on the, um, vertical axis. So in supervised learning,
you are given a dataset with, ah, inputs X and labels Y,
and your goal is to learn a mapping from X to Y.
Right. Now, um, fitting a straight line to data is maybe the simplest possible.
Maybe the simplest possible learning algorithm, maybe one of the simplest poss- learning
algorithms.
Um, given a dataset like this, there are many possible ways to learn a mapping, to learn the
function mapping from the input size to the estimated price.
And so, um, maybe you want to fit a quadratic function instead, maybe that actually fits the
data a little bit better.
And so how do you choose among different models will be, ah, either automatically or
manual intervention will be-
will be something we'll spend a lot time talking about. Now to give a little bit more.

Um, to define a few more things. This example is a problem called a regression problem.
And the term regression refers to that the value y you're trying to predict is continuous.
Right. Um, in contrast, here's a- here's a different type of problem.
Um, so problem that some of my friends were working on, and- and I'll simplify it was- was a
healthcare problem, uh, where,
they were looking at, um, breast cancer or breast tumors, um, and trying to decide if a tumor
is benign or malignant.
Right. So a tumor is a lump in a- in a woman's breast, um, is- can be ma- malign,
or cancerous, um, or benign, meaning you know, roughly it's not that harmful. And so if on
the horizontal axis,
you plot the size of a tumor. Um, and on the vertical axis,
you plot is it malignant or not. Malignant means harmful, right.
Um, and some tumors are harmful some are not. And so whether it is malignant or not,
takes only two values, 1 or 0.
And so you may have a dataset, um, like that.
Right. Ah, and given this, can you learn a mapping from X to Y,
so that if a new patient walks into your office, uh, walks in the doctor's office and the tumor
size is,
you know say, this, can the learning algorithm figure out from this data that it was probably,
well, based on this dataset, looks like there's- there's a high chance that that tumor is, um,
malignant.
Uh, so, uh, so this is an example of
a classification problem and
the term classification refers to that Y here takes on a discrete number of variables.
So for a regression problem, Y is a real number. I guess technically prices can be rounded
off to the nearest dollar and cents,
so prices aren't really real numbers. Um, you know that- because you'd probably not price it,
how's it like Pi times 1 million or whatever. Ah, but, so, so- but for all practical purposes
prices are continuous
so we call the housing price prediction to be a regression problem, whereas if you have,
uh, two values of possible output,
0 and 1, call it a classification problem. Um, if you have K discrete outputs so,
uh, if the tumor can be, uh, malignant or if there are five types of cancer, right,
so you have one of five possible outputs, then that's also a classification problem. If the
output is discrete.
Now, um, I wanna find a different way to visualize this dataset which is,
um, let me draw a line on top. And I'm just going to, you know, map all this data on the
horizontal axis upward onto a line.
But let me show you what I'm gonna do. I'm going to use a symbol O to denote.
Right. Um, I hope what I did was clear. So I took the two sets of examples,
uh, the positive and negative examples. Positive example was this 1, negative example was
0. And I took all of these examples and- and kinda pushed them up onto a straight line,
and used two symbols, used O's to denote negative examples and use crosses to denote
positive examples.
Okay. So this is just a different way of visualizing the same data, um,
by drawing it on the line and using, you know, two symbols to denote the two discrete values
0 and 1, right?
So, um, it turns out that, uh, uh, in both of these examples,
the input X was one-dimensional, it was a single real number. For most of the, um,
machine learning applications you work with, the input X will be multi-dimensional. You won't
be given just one number and asked to predict another number.
Instead, you'll often be given, uh, multiple features and multiple numbers to predict another
number.
So for example, instead of just using a tumor size to predict- to estimate malignancy-
malignant versus benign tumors, um,
you may instead have two features, where one is tumor size and the second is age of the
patient,
and be given a dataset, [NOISE]
right? And be given a dataset that looks like that, right?

Where now your task is, um, given two input features, so X is tumor size and age,
you know, like a two-dimensional vector, um, and your task is given, uh, these two input
features, to predict whether a given tumor is malignant or benign. So if a new patient walks in a
um, uh,
doctor's office
and that the tumor size is here and the age is here, so that point there,
then hopefully you can conclude that, you know, this patient's tumor is probably benign,
right? Corresponding to a O, that negative example.
Um, and so what one thing you'll learn, um, next week is a learning algorithm that can
fit a straight line to the data as follows,
kind of like that, to separate out the positive and negative examples. Separate out the O's and
the crosses.
And so next week, you'll learn about the logistic regression algorithm which, um, which can
do that.
Okay? So, um, one of the most interesting things you'll learn about is, uh, let's see.
So in this example, I drew a dataset with two input features, um,
when- so I have friends that actually worked on the breast cancer, uh, prediction problem,
and in practice you usually have a lot more than one or two features, and usually you have
so many features you can't plot them on the board, right?
And so for an actual breast cancer prediction problem, my friends are working on this were-
were using many other features such as, I guess clump thickness, uh, you know, uniformity
of cell size,
uniformity of cell shape, right? Um, uh, adhesion, how well the cells stick together.
don't worry about what these means,
Don't worry about what this means but, uh, if you are actually doing this in a- in a actual
medical application,
there's a good chance that you'll be using a lot more features than just two. Uh, and this
means that you actually can't plot this data,
right? It's too high-dimensional. You can't plot things higher than 3-dimensional or maybe
4-dimensional, or something like that and so when we have lots of
features it's actually difficult to plot this data. I'll come back to this in a second in learning
theory.
Um, and, uh, one of the things you'll learn about- so as we develop learning algorithms,
you'll learn how to build, um, regression algorithms or classification algorithms that can deal
with these, uh, relatively large number of features.
One of the most fascinating results you'll learn is that, um, [NOISE] you'll also learn about
an algorithm called the Support Vector Machine which uses
not one or two or three or 10 or 100 or a million input features,
but uses an infinite number of input features, right? And so, so, so just to be clear, if in
this example the state of a patient were represents as one number, you know, tumor size, uh,
in this example we had two features. So the state of a patient were represented using two
numbers, the tumor size and the age.
If you use, this list of features maybe a patient that's represented with five or six numbers. Uh,
but there's an algorithm called the support vector machine that
allows you to use an infinite-dimensional vector,
um, to represent a patient. And, um, how do you deal with that
and how can the computer even store an infinite-dimensional vector, right? I mean, you
know, computer memory,
you can store one row number, two row numbers, but you can't store an infinite number of
row numbers in a computer without running out of memory or processor speed or whatever.
So so how do you do that? Uh, so when we talk about support vector machines and
specifically the technical method called kernels,
you'll learn how to build learning algorithms that work with, uh, so that the infinitely long lists
of features,
infinitely long list of feature of- for for- which which- and you can imagine that if you have an
infinitely long list of numbers to represent a patient,
that might give you a lot of information about that patient and so that is one of the relatively
effective learning algorithms to solve problems, okay?
Um, so that's supervised learning. And, you know, let me just,
um, uh, play a video, um, show you a fun- slightly older example
of supervised learning to give you a sense of what this means. [NOISE] But at the heart of
supervised learning is the idea that during training, uh,
you are given inputs X together with the labels Y and you give it both at the same time,
and the job of your learning algorithm is to, uh, find a mapping so that given a new X,
you can map it to the most appropriate output Y. Um, so this is a very old video, uh, made by,
Um,
DA Pomerleau and we've known him for a long time as well, uh, using supervised learning
for autonomous driving. Uh, this is not state of the art for autonomous driving anymore,

but it actually does remarkably well. Oh, and, uh, um, as you, uh, you hear a few technical
terms like back-propagation,
you'll learn all those techniques in this class, uh, and by the end of class, you'll either build a
But let's- let's- let's see this application.
learning algorithm that's much more effective than what you see here.
Uh, could you turn up the volume maybe have that? Are you guys getting volume audio?
[BACKGROUND] Oh, I see.
All right, I'll narrate this. [LAUGHTER] So I'll be using artificial neural network to drive this
vehicle that
uh, was built at Carnegie Mellon University, uh, many years ago. And what happens is, uh,
during training, it watches the human, um, drive the vehicle and I think 10 times per second,
uh, it digitizes the image in front of the vehicle. And, um, so that's a picture taken by a
front-facing camera.
Um, and what it does is in order to collect labeled data, the car while the human is driving it,
records both the image such as it's seeing here, as well as, the steering direction that was
chosen by the human.
So at the bottom here is the image turned to grayscale and lower res, and, uh, on top,
let me pause this for a second. Um, this is the driver direction,
the font's kinda blurry but this text says driver direction. So this is the Y label,
the label Y that the human driver chose. Um, and so the position of this white bar of
this white blob shows how the human is choosing to steer the car. So in this, in this image,
the white blob is a little bit to the left of center so the human is, you know, steering just a little
bit to the left.
Um, this second line here is the output of the neural network and initially,
the neural network doesn't know how to drive, and so it's just outputting this white smear
everywhere and it's saying,
"Nor, I don't know, do I drive left, right, center? I don't know." So it's outputting this gray blur
everywhere.
Um, and as the algorithm learns using the back-propagation learning algorithm
or gradient descents which you'll learn about, uh, you'll actually learn about gradient descent
this Wednesday. Um, you see that
this neural network's output becomes less and less of this white smear, this white blur but
starts to, uh,
becomes sharper, um, and starts to mimic more accurately the human selected driving
direction.
Right. So this, um, is an example of supervised learning because
the human driver demonstrates inputs X and outputs Y, uh, meaning, uh, if you see this in
front of the car steer like that so that's X and Y.
And, uh, after the learning algorithm has learned, um, you can then,
uh, well, he pushes a button, takes the hands off the steering wheel, um, [NOISE] and then
It's using the neural network to drive itself, right?
Digitizing this image in front of the road, taking this image and passing it through the learning
algorithm,
through the trained neural network, letting the neural networks select a steering direction,
Um, and then using a little motor to turn the wheel.
Uh, this is slightly more advanced version which has trained two separate models; one for, I
think, a two-lane road,
one for a four-lane road. Uh, so that's the, um, uh, so the second and third lines this is for a
two-lane road,
this is a four-lane road. And the arbitrator is, is another algorithm that tries to decide whether
the two-lane or the four-lane road model is the more, more appropriate one for a particular
given situation.
Um, and so as Alvin is, excuse me a one-lane road or,
uh, a two-lane road. So, so, so it's driving from a one-lane road here, uh, to another
intersection,
um, the, uh, the the algorithm realizes it should swi- switch over from,
um, I think I forget, I think the one-lane neural network to the- to the two-lane neural network
[NOISE] one.
All right. One of these, right?
Okay. Oh, oh, right. Fine. We'll just see the final dramatic moment of switching from a
[LAUGHTER]
one-lane road to a two-lane road.
All right. Um, uh, and I think, you know, so this is just using supervised learning to- take as
input,

what's in front of the car to decide on the steering direction. This is not state of the art for
but you know, you could do it today,
not self-driving cars and build those things in some limited contexts. Uh, uh, and I think,
uh, in, in several weeks, you'll actually be able to build something that is more sophisticated
than this.
Right. Um, so after supervised learning,
uh, we wi- will- in this class we'll spend a bit of time talking about machine learning strategy.
Also, well, I think on the class notes we annotate this as a learning theory. But what that
means is, um,
giving you the tools to go out and apply learning algorithms effectively. And I think I've
been fortunate to know,
uh, you know, to have a lot of, uh, uh, I, I think that,
um, I've been fortunate to have, you know, over the years constantly visited lots of great tech
companies.
Uh, um, more than once that I've that- that I've been probably associated with, right? But
often, just to help friends out,
I visit various tech companies, uh, whose products I'm sure are installed on your cell phone.
Uh, but I often visit tech companies and you know, talk to the machine learning teams and
see what they're doing, and see if I can help them out.
And what I see is that, there's a huge difference in the effectiveness of how
two different teams could apply the exact same learning algorithm. All right? Uh, and I think
that, um,
what I've seen sadly is that sometimes there will be a team or even in some of the best tech
companies, right?
The, the, the EV, AI companies, right? And, and, and multiple of them, where you go talk
to a team and they'll
tell you about something that they've been working on for six months. And then, you can
quickly take a look at the data and,
and hear that they're not- the algorithm isn't quite working and sometimes you can look at
what they're doing,
and go yeah, you know, I could have told you six months ago that this approach is never
gonna work, right?
Um, and, um, what I find is that the most skilled machine learning practitioners are very
strategic.
By which I mean that your skill at deciding- um, when you work on a machine learning
project,
you have- you- you have a lot of decisions to make. Right? Do you collect more data? Do
you try a different learning algorithm?
Uh, do you buy a faster GPU to train your learning algorithm for longer? Or if you collect
more data, what type of data do you collect?
Or for all of these architectural choices, using neural networks for reference machine which
Is regression, which ones do you pick?
Um, but there are a lot of decisions you need to make when building these learning
algorithms.
So one thing that's quite unique to the way we teach is, uh, we want to help you become
more systematic in driving machine learning as a,
as a systematic engineering discipline, so that when one day when you are working on as
machine learning project, you can efficiently figure out what to do next.
Right? Um, and I sometimes make an analogy to how, um, to, uh, uh,
to, to software engineering. Um, you know, like many years ago, I had a friend, um,
that would debug code by compiling it and then, um, uh, this friend would look for all of the
syntax errors, right?
That, you know, C++ compiler outputs. And they thought that the best way to eliminate the
errors
is to delete all the lines of code with syntax errors and that was [LAUGHTER] their first
heuristic. So that did not go well, right?
Uh, but it took me a while to persuade them to stop doing that. Uh, but, but, but so it turns
out that, um,
when you run a learning algorithm, you know, it almost never works the first time. All right?
It's, this, just life.
Uh, and the way you go about debugging the learning algorithm will have a huge
impact on your efficiency, um, on,
on how quickly you can build effective learning systems. And I think until now, too much of
the- of this process of, uh,
making your learning algorithms work well has been a black magic kind of process where,
you have worked on this for decades.
So when you run something, you don't know why it does not recognize it like, "Hey, what do I
do about it, and then, because he's
And then says, "Oh, yeah, do that." so experienced it works, but I think, um, what we're trying
is to evolve the discipline of machine learning
to do with it from a black magic, tribal knowledge, experience-based thing to a systematic
engineering process.
All right. And so, um, later this quarter, as we talk about machine learning strategy,
we'll talk about learning theory. We'll try to systematically give you tools on how to, um, uh,
go about strategizing.

Uh, so- so it can be very efficient in, um, how you- how you yourself, how you can lead a
team to build an effective learning system,
because I don't want you to be one of those people that, you know, wastes six months on
some direction that
maybe could have relatively quickly figured out it was not promising. Or maybe one last
analogy,
um, if you're used to optimizing code, right? Making code run faster,
I'm not sure if you have done that. Uh, uh, uh, less experienced software engineers,
who'll just dive in and optimize the code, they try to make it run faster, right? Let's take the
C++ code and code in assembly or something.
But more experienced people will run a profiler to try to figure out what part of code is
actually the bottleneck and then just focus on changing that.
So, uh, one of the things we hope to do this quarter is, uh, uh, convey to you some of these
more systematic engineering principles.
All right. And yeah. Oh, and actually this is very interesting. This is a, uh, uh, yeah. Actually,
I've actually- I've been invited,
so a bunch of- so how many of you have heard the machine learning journey? Oh, just a few of
you, interesting.
Oh, so actually, so, so- if any of you are interested, um, just in my, uh, spare time, uh,
I've been writing a book, um, uh, to try to codify systematic engineering principles for
machine learning and so if you, uh,
uh, and so um, if you want a, you know, free draft copy of the book, sign up for a mailing list
here.
I tend to just write stuff and put it on the Internet for free, yeah. So if you want a free draft
copy of the book,
uh, you know, go to this website, uh, enter your e-mail address and the website will send
you a copy of the book.
And I'll talk a little bit about these engineering principles as well. Okay. All right. So, uh,
so first subject, machine learning. Second subject, learning theory. Um, and, uh, the third
major subject we'll talk about is, uh, deep learning, right?
And so you have a lot of tools in machine learning and many of them are worth learning
about and, for many different applications.
you know, use in many different applications. There is one subset of machine learning that's
really hot right now because it's just advancing very rapidly, so that you can understand the
And so we'll spend a little time talking about deep learning,
basics of how to train a neural network as well.
But I think that CS229 covers a much broader set of algorithms which are all useful.
CS230, more narrowly covers just deep learning, right? Um.
So, uh, other than deep learning slash after- after deep learning slash neu- neural networks
the other,
four of the five major topics we'll cover will be on unsupervised learning.
Um, so what is unsupervised learning? [NOISE]
So you saw me draw a picture like this just now, right?
And this would be a classification problem like the tumor, malignant, benign problems, this is
a classification problem.
And that was a supervised learning problem because you have to learn a function mapping
from X to Y.
Um, unsupervised learning would be if I give you a dataset like this with no labels.
So you're just given inputs X and no Y, and you're asked to find me something interesting in
this data,
figure out, you know, interesting structure in this data. Um, and so in this dataset, it looks like
there are two clusters,
and the unsupervised learning algorithm which we learned about called K-means
this structure in the data. Um,
clustering, will discover this, um, other examples as well as learning, you know,
if- if you actually, Google News is a very interesting website. Sometimes I use it to look up,
right?
latest news, just this old example. But Google News everyday crawls or reads,
uh, uh, I don't know, uh, uh, many many thousands or tens of thousands of news articles on
the internet and groups them together, right?
For example, there's a set of articles on the BP Oil Well spill, and it has,
uh, taken a lot of the articles written by different reporters and grouped them together.
So you can, you know, figure out that what BP, uh, Macondo oil well, right?
That this is a CNN article about the oil well spill, there's a Guardian article about oil well spill
and this is
an example of a clustering algorithm whereas taking these different news sources and
figuring out that these are
all stories kind of about the same thing, right? Um, and other examples of clustering,

just getting data and figuring out what groups belong together. Um, a lot of work on, um, genetic data.
This is a visualization of- of genetic microarray data.
Where given data like this, you can group individuals into different types of- into individuals of different characteristics, um,
clustering algorithms grouping this type of data together is used to, um,
organize computing clusters, you know, figure out what machines workflows are more related to each other and organize computing clusters appropriately.
So take a social network like LinkedIn or Facebook or other social networks and figure out which are the groups of
friends and which are the cohesive communities within a social network, um, or market segmentation.
Um, actually many companies I've worked with look at the customer database and cluster the users together, um. So you can say, uh, that looks like we have four types of users, you know,
looks like there are the, uh, young professionals looking to develop themselves,
there are the, you know, soccer moms and soccer dads, there are this category and these category- and you can then market to the different market segments, um, separately.
Uh, and- and actually many years ago my friend Andrew Moore, uh, uh,
was using this type of data for astronomical data analysis to group together galaxies. You have a question?
[inaudible] Is unsupervised learning the same as learning clustering? No it's not.
So unsupervised learning broadly is the concept of using unlabeled data. So just X and finding interesting things about it, right?
Um, so, um, for example, uh, actually here's- shoot.
This won't work with all of you will do this later in the cla- in the class I guess. Um, maybe I say we'll do this later.
Cocktail party problem, um, uh, is another unsupervised learning problem.
Reading the audio for this to explain this though, um, let me think how to explain this.
Um, the cocktail party problem and I'll try to do the demo when we can get all your work on this laptop.
Is a problem where, um, if you have a noisy room and you stick mult- multiple microphones in the room and record overlapping voices,
um, so that you have multiple microphones, an array of microphones, in a room of lots of people talking.
Uh, how can you have the algorithm separate out the people's voices. So that's an unsupervised learning problem because,
um, there are no labels. So you just stick microphones in the room and have it record you have multiple voices, overlapping voices,
different people's voices at the same time and then have it try to separate out people's voices. And one of the programming exercises you do later is,
if you have, you know, five people talking. So each microphone records five people's overlapping voices, right?
Because, you know, each microphone hears five people at the same time. How can you have an algorithm separate out
these voices so you get clean recordings of just one voice at a time. So that's called the cocktail party problem and the algorithm you
use to do this is called ICA, Independent Components Analysis. And that's something you implement in one of the later homework exercises, right?
Um, and there are other examples of unsupervised learning as well. Uh, the Internet has tons of unlabeled text data.
You just suck down data from the Internet. There are no labels necessarily but can you learn interesting things about language,
figure out what- figure out, um, I don't know, one of the best cited results recently was learning analogies,
like, yeah, man is to woman as king is to queen, right? Or a- what's a Tokyo is to Japan as Washington DC is to the United States, right?
To learn analogies like that. It turns out you can learn analogies like that from unlabeled data, just from text on the Internet.
So that's also unsupervised learning. Okay? Um, so after unsupervised learning,
oh, and unsupervised learning. So you know machine learning is very useful today. It turns out that most of the recent wave of
economic value created by machine learning is through supervised learning. Uh, but there are important use cases for unsupervised learning as well.
So I use them in my work occasionally. Uh, and is also a bleeding edge for a lot of exciting research.
And then the final topic. Finally, the five topics we covered. So talk about supervised deep learning, machine learning strategy,
learning, unsupervised learning, and then the fifth one is reinforcement learning, is this. Which is, um, let's say, I give you the keys to Stanford autonomous helicopter.
This helicopter is actually sitting in my office, and I'm trying to figure out how to get rid of it. Um, and I'll ask you to write a program to make it fly, right?
So how do you do that? Um, so this is a video of a helicopter flying.
The audio is just a lot of helicopter noise. So that's not important. But we'll zoom out the video.

You see she's found in the sky, right? There. So, um, you can use learning alg- that's kinda
cool, right?
[LAUGHTER] I was- I was the camera man that day. Um, but so you can use learning
algorithms to get
you know, robots, to do pretty interesting things like this. Um, and it turns out that a good way
to do this is through reinforcement learning.
So what's reinforcement learning? Um, it turns out that no one knows what's the optimal way
to fly a helicopter, right?
If you fly a helicopter, you have two control sticks that you're moving. Um, but no one knows
what's the optimal way to move the control stick.
So the way you can get a helicopter to fly itself is, um, let the helicopter do whatever- think of
this as training a dog, right?
You can't teach a dog the optimal way to behave, but- actually, how many of you have had a
pet dog or a pet cat before?
Oh, not that many of you? This is fascinating. Okay. So I had a pet dog when I was a kid
and my family made it my job to train the dog. So how do you train a dog? You let the dog do
whatever it wants,
and then whenever it behaves well, you go, "Oh, good dog". And when it misbehaves you go,
"bad dog".
[LAUGHTER] Um, and then over time, the dog learns to do more of the good dog things and
fewer of the bad dog things,
and so reinforcement learning is a bit like that, right? I don't know what's the optimal way to
fly a helicopter. So you let the helicopter do whatever it wants and then whenever it flies well,
does some maneuver you want, or flies accurately without jittering around too much, you go,
"Oh, good helicopter".
[LAUGHTER] And when it crashes you go, "bad helicopter" and it's the job of the
reinforcement learning algorithms to figure out how to
control it over time so as to get more of the good helicopter things and fewer of the bad
helicopter things.
Um, and I think, um, well, just one more video. Um, oh, yeah, that's interesting. All right.
And so again given a robot like this, I actually don't know how to program a- actually a robot
like this has a lot of joints, right?
So how do you get a robot like this to climb over obstacles? So well, this is actually a robot
dog,
so you can actually say, "Good dog" or "Bad dog". [LAUGHTER] By giving those signals,
called a reward signal, uh, you can have a learning algorithm figure out by itself, how to
optimize the reward,
and therefore, [LAUGHTER] climb over these types of obstacles. Um, and I think recently,
the most famous applications of reinforcement learning happened for game-playing, playing
Atari games or playing Go,
you know, Game of Go, like AlphaGo. I think that's a- I think that is a- game playing has
made some remarkable strides
remarkable PR but I'm also equally excited or maybe even more excited about the integrals
into reinforcement learning making
and robotics applications, right? So I think, um, I think- yeah, reinforcement learning has
been proven to be fantastic at playing
games, it's also getting real traction in optimizing robots and optimizing sort of
logistic systems and things like that.
Um, so you learned about all these things. Um, last thing for today, uh,
I hope that you will start to, to meet people in the class, make friends, find project partners
and study groups,
if you have any questions, [NOISE] you know, dive on the Piazza, asking questions as
you help each other answer their questions.
So let's break for today and I look forward to seeing you on Wednesday. Welcome to 229.