'MachineLearning-Lecture01 nInstructor (Andrew Ng): Okay. Good morning. Welcome to CS229, the machine nlearning class. So what I wanna do today is ju st spend a little time going over the logistics nof the class, and then we'll start to talk a bit about machine learning. nBy way of introduction, my name's Andrew Ng and I'll be instru ctor for this class. And so nI personally work in machine learning, and I' ve worked on it for about 15 years now, and nI actually think that machine learning is th e most exciting field of all the computer nsciences. So I'm actually always excited about teaching this class. Sometimes I actually nthink that machine learning is not only the most exciting thin g in computer science, but nthe most exciting thing in all of human e ndeavor, so maybe a little bias there. nI also want to introduce the TAs, who are all graduate students doing research in or nrelated to the machine learni ng and all aspects of machin e learning. Paul Baumstarck nworks in machine learning and computer vision. Catie Chang is actually a neuroscientist nwho applies machine learning algorithms to try to understand the human brain. Tom Do nis another PhD student, works in computa tional biology and in sort of the basic nfundamentals of human learning. Zico Kolter is the head TA — he's head TA two years nin a row now — works in machine learning a nd applies them to a bunch of robots. And nDaniel Ramage is — I guess he's not here — Daniel applies l earning algorithms to nproblems in natural language processing. nSo you'll get to know the TAs and me much be tter throughout this quarter, but just from nthe sorts of things the TA's do, I hope you can already tell that machine learning is a nhighly interdisciplinary topic in which just the TAs find l earning algorithms to problems nin computer vision and biology and robots a nd language. And machine learning is one of nthose things that has and is having a large impact on many applications. nSo just in my own daily work, I actually frequently end up talking to people like nhelicopter pilots to biologists to people in computer systems or databases to economists nand sort of also an unending stream of people from industry coming to Stanford ninterested in applying machine learni ng methods to their own problems. nSo yeah, this is fun. A couple of weeks ago, a student actually forwar ded to me an article nin "Computer World" about the 12 IT skills th at employers can't say no to. So it's about nsort of the 12 most desirabl e skills in all of IT and all of information technology, and ntopping the list was actually machine lear ning. So I think this is a good time to be nlearning this stuff and learning algorithms and having a large impact on many segments nof science and industry. nI'm actually curious about something. Learni ng algorithms is one of the things that ntouches many areas of science and industrie s, and I'm just kind of curious. How many npeople here are computer science majors, are in the computer science department? Okay. nAbout half of you. How many people are from EE? Oh, okay, maybe about a fifth. How '"many biologers are there here? Wow, just a few, not many. I'm surprised. Anyone from nstatistics? Okay, a few. So where are the rest of you from? nStudent : iCME. nInstructor (Andrew Ng) : Say again? nStudent : iCME. nInstructor (Andrew Ng) : iCME. Cool. nStudent : [Inaudible]. nInstructor (Andrew Ng) : Civi and what else? nStudent : [Inaudible] nInstructor (Andrew Ng) : Synthesis, [inaudible] systems. Yeah, cool. nStudent : Chemi. nInstructor (Andrew Ng) : Chemi. Cool. nStudent : [Inaudible]. nInstructor (Andrew Ng) : Aero/astro. Yes, right. Yeah, okay, cool. Anyone else? nStudent : [Inaudible]. nInstructor (Andrew Ng) : Pardon? MSNE. All ri ght. Cool. Yeah. nStudent : [Inaudible]. nInstructor (Andrew Ng) : Pardon? nStudent : [Inaudible]. nInstructor (Andrew Ng) : Endo — nStudent : [Inaudible]. nInstructor (Andrew Ng) : Oh, I see, industry. Okay. Cool. Great, great. So as you can ntell from a cross-section of th is class, I think we're a very diverse audience in this room, nand that's one of the things that makes this class fun to teach and fun to be in, I think. ""So in this class, we've tried to convey to you a broad set of principl es and tools that will nbe useful for doing many, many things. And ev ery time I teach this class, I can actually nvery confidently say that af ter December, no matter what yo u're going to do after this nDecember when you've sort of completed this class, you'll find the things you learn in nthis class very useful, and these things will be useful pretty much no matter what you end nup doing later in your life. nSo I have more logistics to go over later, but let's say a few more words about machine

nlearning. I feel that machine learning grew out of early work in AI, early work in artificial nintelligence. And over the last — I wanna say last 15 or last 20 years or so, it's been viewed as a sort of growing new capability for computers. And in particular, it turns out nthat there are many programs or there are many applications that you can't program by nhand. nFor example, if you want to get a computer to read handwritten characters, to read sort of nhandwritten digits, that actual ly turns out to be amazingly difficult to write a piece of nsoftware to take this input, an image of some thing that I wrote and to figure out just what nit is, to translate my cursive handwriting into — to extract the characters I wrote out in nlonghand. And other things: One thing that my students and I do is autonomous flight. It nturns out to be extremely difficult to sit dow n and write a program to fly a helicopter. nBut in contrast, if you want to do things like to get software to fl y a helicopter or have nsoftware recognize handwritten digits, one very successful approach is to use a learning nalgorithm and have a computer learn by its elf how to, say, recognize your handwriting. nAnd in fact, handwritten digit recognition, this is pretty much the only approach that nworks well. It uses applications that are hard to program by hand. nLearning algorithms has also made I guess sign ificant inroads in what's sometimes called ndatabase mining. So, for example, with the growth of IT and computers, increasingly nmany hospitals are keeping around medical reco rds of what sort of patients, what nproblems they had, what their prognoses was, what the outcome was. And taking all of nthese medical records, which started to be digitized only about maybe 15 years, applying nlearning algorithms to them can turn raw medi cal records into what I might loosely call nmedical knowledge in which we start to detect trends in medical practice and even start to nalter medical practice as a result of me dical knowledge that's derived by applying nlearning algorithms to the sorts of medical r ecords that hospitals have just been building nover the last 15, 20 years in an electronic format. nTurns out that most of you probably use learning algorithms — I don't know — I think nhalf a dozen times a day or maybe a dozen times a day or more, and often without nknowing it. So, for example, every time you se nd mail via the US Postal System, turns nout there's an algorithm that tries to automa tically read the zip code you wrote on your nenvelope, and that's done by a learning al gorithm. So every time you send US mail, you nare using a learning algorithm, perhap s without even being aware of it. ""Similarly, every time you write a check, I ac tually don't know the number for this, but a nsignificant fraction of checks that you write are processed by a learning algorithm that's nlearned to read the digits, so the dolla r amount that you wrote down on your check. So nevery time you write a check, there's anot her learning algorithm that you're probably nusing without even being aware of it. nIf you use a credit card, or I know at least one phone compan y was doing this, and lots of ncompanies like eBay as well that do electr onic transactions, there's a good chance that nthere's a learning algorithm in the backgr ound trying to figure out if, say, your credit ncard's been stolen or if someone's engaging in a fraudulent transaction. nIf you use a website like Amazon or Netflix that will often recommend books for you to nbuy or movies for you to rent or whatever , these are other examples of learning nalgorithms that have learned what sorts of th ings you like to buy or what sorts of movies nyou like to watch and can therefore give customized recommendations to you. nJust about a week ago, I had my car serviced, and even there, my car mechanic was trying nto explain to me some learning algorithm in th e innards of my car th at's sort of doing its nbest to optimize my driving performan ce for fuel efficiency or something. nSo, see, most of us use learning algorithms half a dozen, a dozen, maybe dozens of times nwithout even knowing it. nAnd of course, learning algorithms are also doing things like giving us a growing nunderstanding of the human genome. So if so meday we ever find a cure for cancer, I bet nlearning algorithms will have had a large role in that. That's sort of the thing that Tom nworks on, yes? nSo in teaching this class, I sort of have thre e goals. One of them is just to I hope convey nsome of my own excitement a bout machine learning to you. nThe second goal is by the end of this class, I hope all of you will be ab le to apply state-of-nthe-art machine learning algorithms to whatev er problems you're

interested in. And if you never need to build a system for reading zi p codes, you'll know how to do that by the end nof this class. nAnd lastly, by the end of this class, I reali ze that only a subset of you are interested in ndoing research in machine learning, but by the c onclusion of this class, I hope that all of nyou will actually be well qualified to star t doing research in machine learning, okay? nSo let's say a few words about logistics. The prerequisites of this class are written on one nof the handouts, are as follows: In this class, I'm going to assume that all of you have sort nof basic knowledge of computer science and kn owledge of the basic computer skills and nprinciples. So I assume all of you know what big?O notation, that all of you know about nsort of data structures like queues, stacks, binary trees , and that all of you know enough nprogramming skills to, like, write a simple co mputer program. And it turns out that most ""of this class will not be very program ming intensive, although we will do some nprogramming, mostly in either MATLAB or Octa ve. I'll say a bit more about that later. nI also assume familiarity with basic proba bility and statistics. So most undergraduate nstatistics class, like Stat 116 taught here at Stanford, will be more than enough. I'm gonna nassume all of you know what ra ndom variables are, that all of you know what expectation nis, what a variance or a random variable is. And in case of some of you, it's been a while nsince you've seen some of this material. At some of the discussion sections, we'll actually ngo over some of the prerequisites, sort of as a refresher course under prerequisite class. nI'll say a bit more about that later as well. nLastly, I also assume familiarity with basi c linear algebra. And again, most undergraduate nlinear algebra courses are more than enough. So if you've taken courses like Math 51, n103, Math 113 or CS205 at Stanford, that would be more than enough. Basically, I'm ngonna assume that all of you know what matrix es and vectors are, that you know how to nmultiply matrices and vectors and multiply matrix and matrices, that you know what a matrix inverse is. If you know what an eigenvect or of a matrix is, that'd be even better. nBut if you don't quite know or if you're not qu ite sure, that's fine, too. We'll go over it in nthe review sections. nSo there are a couple more logisti cal things I should deal with in this class. One is that, as nmost of you know, CS229 is a televised cla ss. And in fact, I guess many of you are nprobably watching this at home on TV, so I' m gonna say hi to our home viewers. nSo earlier this year, I appro ached SCPD, which televises th ese classes, about trying to nmake a small number of Stanford classes publ icly available or posting the videos on the nweb. And so this year, Stanford is actually starting a small pilot program in which we'll npost videos of a small number of classes onlin e, so on the Internet in a way that makes it npublicly accessible to everyone. I'm very exc ited about that because machine learning in nschool, let's get the word out there. nOne of the consequences of this is that — let's see — so videos or pictures of the students nin this classroom will not be posted online, so your images — so don't worry about being nby seeing your own face appear on YouTube one day. But the microphones may pick up your voices, so I guess the consequence of that is that because microphones may pick up your voices, no matter how irritated you are at me, don't yell out swear words in the nmiddle of class, but because there won't be video you can safely sit there and make faces nat me, and that won't show, okay? nLet's see. I also handed out this — ther e were two handouts I hope most of you have, ncourse information handout. So let me just sa y a few words about parts of these. On the nthird page, there's a section that says Online Resources. nOh, okay. Louder? Actually, could you turn up the volume? Testing. Is this better? nTesting, testing. Okay, cool. Thanks. ""So all right, online resources. The class has a home page, so it's in on the handouts. I nwon't write on the chalkboard — http:// cs229.stanford.edu. And so when there are nhomework assignments or things like that, we usually won't sort of — in the mission of nsaving trees, we will usually not give out many handouts in class. So homework nassignments, homework solutions will be posted online at the course home page. nAs far as this class, I've also written, a nd I guess I've also revised every year a set of nfairly detailed lecture notes that cover the te chnical content of this class. And so if you nvisit the course homepage, you'll also find the detailed lecture notes that go over in detail nall the math and

equations and so on that I'll be doing in class. nThere's also a newsgroup, su.class.cs229, also written on the handout. This is a nnewsgroup that's sort of a forum for people in the class to get to know each other and nhave whatever discussions you want to ha ve amongst yourselves. So the class newsgroup nwill not be monitored by the TAs and me. But this is a place for you to form study groups nor find project partners or discuss homework problems and so on, and it's not monitored nby the TAs and me. So feel free to ta lk trash about this class there. nIf you want to contact the teaching staff, pl ease use the email address written down here, ncs229-qa@cs.stanford.edu. This goes to an acc ount that's read by all the TAs and me. So nrather than sending us email individually, if you send email to this account, it will nactually let us get back to you maximally quickly with answers to your questions. nIf you're asking questions about homework probl ems, please say in the subject line which nassignment and which question the email refers to, since that will also help us to route nyour question to the appropriate TA or to me appropriately and get the response back to nyou quickly. nLet's see. Skipping ahead — let's see — for homework, one midterm, one open and term nproject. Notice on the honor code. So one thi ng that I think will help you to succeed and ndo well in this class and even help you to enjoy this cla ss more is if you form a study ngroup. nSo start looking around where you' re sitting now or at the end of class today, mingle a nlittle bit and get to know your classmates. I strongly encourage you to form study groups nand sort of have a group of people to study with and have a group of your fellow students nto talk over these concepts with. You can also post on the class news group if you want to nuse that to try to form a study group. nBut some of the problems sets in this cla ss are reasonably difficult. People that have ntaken the class before may tell you they were very difficult. And just I bet it would be nmore fun for you, and you'd probably have a be tter learning experience if you form a nstudy group of people to work with. So I definitely encourage you to do that. nAnd just to say a word on the honor code, whic h is I definitely en courage you to form a nstudy group and work together, discuss homew ork problems together. But if you discuss ""homework problems with other students, then I'll ask you to sort of go home and write ndown your own solutions independe ntly without referring to note s that were taken in any nof your joint study sessions. nSo in other words, when you turn in a hom ework problem, what you turn in should be nsomething that was reconstructed independe ntly by yourself and w ithout referring to nnotes that you took during your study sessions with other people, okay? And obviously, nshowing your solutions to othe rs or copying other solutions directly is right out. nWe occasionally also reuse problem set questions from previous years so that the nproblems are a bit more debugged and work more smoothly. And as a result of that, I also nask you not to look at solutions from previous ye ars, and this includes both sort of official nsolutions that we've given out to previous gene rations of this class and previous solutions nthat people that have taken this class in previous years may have written out by nthemselves, okay? nSadly, in this class, there are usually — sadly, in previous y ears, there have often been a nfew honor code violations in this class. And last year, I think I pr osecuted five honor code nviolations, which I think is a ridiculously large number. And so just don't work without nsolutions, and hopefully there'll be zero honor code violations this year. I'd love for that nto happen. nThe section here on the late homework polic y if you ever want to hand in a homework nlate, I'll leave you to r ead that yourself. nWe also have a midterm, which is scheduled for Thursday, 8th of November at 6:00 p.m., nso please keep that evening free. nAnd let's see. And one more administrative thing I wanted to sa y is about the class nproject. So part of the goal of this cla ss is to leave you well eq uipped to apply machine nlearning algorithms to a problem or to do rese arch in machine learning. And so as part of nthis class, I'll ask you to execute a small resear ch project sort of as a small term project. nAnd what most students do for this is either apply machine learning to a problem that you nfind interesting or investigate some aspect of machine learning. So to those of you that nare either already doing research or to those of you who are in industry, you're taking this nfrom a company, one fantastic sort of way to do a class project would be if you apply nmachine

learning algorithms to a problem that you're interested in, to a problem that nyou're already working on, whether it be a scien ce research problem or sort of a problem nin industry where you're trying to get a syst em to work using a learning algorithm. nTo those of you that are not currently doing re search, one great way to do a project would nbe if you apply learning algorithms to just pick a problem that you care about. Pick a nproblem that you find interesting, and apply lear ning algorithms to that and play with the nideas and see what happens. ""And let's see. Oh, and the goal of the projec t should really be for you to do a publishable npiece of research in machine learning, okay? nAnd if you go to the course website, you'll actuall y find a list of the projects that students nhad done last year. And so I'm holding the li st in my hand. You can go home later and ntake a look at it online. nBut reading down this list, I see that last year, there were st udents that ap plied learning nalgorithms to control a snake robot. Ther e was a few projects on improving learning nalgorithms. There's a project on flying autonomous aircraft. There was a project actually ndone by our TA Paul on improvi ng computer vision algorithms using machine learning. nThere are a couple of project s on Netflix rankings using learning algorithms; a few nmedical robots; ones on segmenting [inaudibl e] to segmenting pieces of the body using nlearning algorithms; one on musical instrume nt detection; anot her on irony sequence nalignment; and a few algorithms on understandin g the brain neuroscience, actually quite a nfew projects on neuroscience; a couple of projects on unde scending fMRI data on brain nscans, and so on; another project on market makings, the financial trading. There was an ninteresting project on trying to use learning algorithms to decide what is it that makes a nperson's face physically attractive. There's a learning algorithm on op tical illusions, and nso on. nAnd it goes on, so lots of fun projects. A nd take a look, then come up with your own nideas. But whatever you find cool and interest ing, I hope you'll be able to make machine nlearning a project out of it. Yeah, question? nStudent : Are these gro up projects? nInstructor (Andrew Ng): Oh, yes, thank you. nStudent : So how many people can be in a group? nInstructor (Andrew Ng): Right. So projects can be done in groups of up to three people. nSo as part of forming study groups, later t oday as you get to know your classmates, I ndefinitely also encourage you to grab two ot her people and form a group of up to three npeople for your project, okay? And just start brainstorming ideas for now amongst nyourselves. You can also come and talk to me or the TAs if you want to brainstorm ideas nwith us. nOkay. So one more organizational ques tion. I'm curious, how many of you know nMATLAB? Wow, cool, quite a lot. Okay. So as part of the — act ually how many of you nknow Octave or have used Octave ? Oh, okay, much smaller number. nSo as part of this class, especially in the homeworks, we'll ask you to implement a few nprograms, a few machine learning algorithms as part of the homeworks. And most of "'those homeworks will be done in either MATLA B or in Octave, which is sort of — I nknow some people call it a free ve rsion of MATLAB, which it sort of is, sort of isn't. nSo I guess for those of you that haven't s een MATLAB before, and I know most of you nhave, MATLAB is I guess part of the programming language that makes it very easy to write codes using matrices, to write code for numerical routines, to move data around, to nplot data. And it's sort of an extremely easy to learn tool to use for implementing a lot of nlearning algorithms. nAnd in case some of you want to work on your own home computer or something if you ndon't have a MATLAB license, for the purposes of this class, there's also — [inaudible] nwrite that down [inaudible] MATLAB — there' s also a software package called Octave nthat you can download for free off the Internet. And it has somewhat fewer features than MATLAB, but it's free, and for the purposes of this class, it will work for just about neverything. nSo actually I, well, so yeah, just a side comment for those of you that haven't seen nMATLAB before I guess, once a colleague of mine at a different university, not at nStanford, actually teaches another machine l earning course. He's taught it for many years. nSo one day, he was in his office, and an old student of his from, lik e, ten years ago came ninto his office and he said, "Oh, professo r, professor, thank you so much for your nmachine learning class. I learned so much from it. There's this stuff that I learned in your nclass, and I now use every

day. And it's help ed me make lots of money, and here's a npicture of my big house." nSo my friend was very excited. He said, "W ow. That's great. I'm glad to hear this nmachine learning stuff was actually useful. So what was it that you learned? Was it nlogistic regression? Was it the PCA? Was it the data ne tworks? What was it that you nlearned that was so helpful?" And the student said, "Oh, it was the MATLAB." nSo for those of you that don't know MATLAB yet, I hope you do learn it. It's not hard, nand we'll actually have a short MATLAB tutori al in one of the discussion sections for nthose of you that don't know it. nOkay. The very last piece of logistical th ing is the discussion s ections. So discussion nsections will be taught by the TAs, and atte ndance at discussion sections is optional, nalthough they'll also be recorded and televi sed. And we'll use the discussion sections nmainly for two things. For the next two or th ree weeks, we'll use the discussion sections nto go over the prerequisites to this class or if some of you haven't seen probability or nstatistics for a while or maybe algebra, we'll go over those in the discussion sections as a nrefresher for those of you that want one. nLater in this quarter, we'll also use the disc ussion sections to go over extensions for the nmaterial that I'm teaching in the main lectur es. So machine learning is a huge field, and nthere are a few extensions that we really want to teach but didn't have time in the main nlectures for. '"So later this quarter, we'll use the discussion sections to talk about things like convex noptimization, to talk a little bit about hidde n Markov models, which is a type of machine nlearning algorithm for modeling time series and a few other things, so extensions to the nmaterials that I'll be covering in the main lectures. And attend ance at the discussion nsections is optional, okay? nSo that was all I had from l ogistics. Before we move on to start talking a bit about nmachine learning, let me check what questions you have. Yeah? nStudent : [Inaudible] R or something like that? nInstructor (Andrew Ng) : Oh, yeah, let's see, right. So our policy has been that you're nwelcome to use R, but I would strongly advi se against it, mainly because in the last nproblem set, we actually supply some code th at will run in Octave but that would be nsomewhat painful for you to translate into R yourself. So for your other assignments, if nyou wanna submit a solution in R, that's fi ne. But I think MATLAB is actually totally nworth learning. I know R and MATLAB, and I personally end up using MATLAB quite a nbit more often for various reasons. Yeah? nStudent : For the [inaudible] pr oject [inaudible]? nInstructor (Andrew Ng) : So for the term project, you're welcome to do it in smaller ngroups of three, or you're welcome to do it by yo urself or in groups of two. Grading is the nsame regardless of the group size, so with a larger group, you probably — I recommend ntrying to form a team, but it's actually totally fine to do it in a sma ller group if you want. nStudent : [Inaudible] what language [inaudible]? nInstructor (Andrew Ng): So let's see. There is no C programming in this class other nthan any that you may choose to do yourself in your project. So all the homeworks can be ndone in MATLAB or Octave, and let's see. A nd I guess the program prerequisites is more nthe ability to understand big?O notation and know ledge of what a data structure, like a nlinked list or a queue or bina ry treatments, more so than your knowledge of C or Java nspecifically. Yeah? nStudent : Looking at the end semester project, I mean, what exactly will you be testing nover there? [Inaudible]? nInstructor (Andrew Ng) : Of the project? nStudent : Yeah. nInstructor (Andrew Ng) : Yeah, let me answer that later. In a couple of weeks, I shall ngive out a handout with guidelines for the pr oject. But for now, we should think of the ngoal as being to do a cool piec e of machine learning work that will let you experience the "'joys of machine learning firs thand and really try to think about doing a publishable piece nof work. nSo many students will try to build a cool machine learning application. That's probably nthe most common project. Some students will try to improve state-of-the-art machine nlearning. Some of those projects are also very successful. It's a littl e bit harder to do. And nthere's also a smaller minority of students th at will sometimes try to prove — develop the ntheory of machine learning further or try to prove theorems about machine learning. So nthey're usually great projects of all of those types with applications and machine learning nbeing the most common. Anything else? Okay, cool. nSo that was it for logistics. Let's talk about learning algorithms. So can I

have the laptop ndisplay, please, or the projector? Actually, co uld you lower the big sc reen? Cool. This is namazing customer service. Thank you. I see. Okay, cool. Okay. No, that's fine. I see. nOkay. That's cool. Thanks. Okay. nBig screen isn't working toda y, but I hope you can read things on the smaller screens out nthere. Actually, [inaudible] I think this room just got a new projector that — someone nsent you an excited email — was it just on Frid ay? — saying we just got a new projector nand they said 4,000-to-1 something or othe r brightness ratio. I don't know. Someone was nvery excited about the new projector in this room, but I guess we'll see that in operation non Wednesday. nSo start by talking about what machine learni ng is. What is machine learning? Actually, ncan you read the text out there? Raise your hand if the text on the small screens is legible. nOh, okay, cool, mostly legible. Okay. So I'll just read it out. nSo what is machine learning? Way back in about 1959, Arthur Samuel defined machine nlearning informally as the [inaudible] that gives computers to learn — [inaudible] that ngives computers the ability to learn without being explicitly programmed. So Arthur nSamuel, so way back in the history of m achine learning, actually did something very ncool, which was he wrote a checkers progr am, which would play games of checkers nagainst itself. nAnd so because a computer can play thousands of games against itself relatively quickly, nArthur Samuel had his program play thousands of games against itself, and over time it nwould start to learn to rec ognize patterns which led to wi ns and patterns which led to nlosses. So over time it learned things like that , "Gee, if I get a lot of pieces taken by the nopponent, then I'm more likely to lose than win," or, "Gee, if I get my pieces into a ncertain position, then I'm especially li kely to win rather than lose." nAnd so over time, Arthur Samuel had a check ers program that woul d actually learn to nplay checkers by learning what are the sort of board positions that tend to be associated nwith wins and what are the boa rd positions that tend to be associated with losses. And nway back around 1959, the amazing thing about this was that his program actually nlearned to play checkers much better than Arthur Samuel himself could. '"So even today, there are some people that say, well, computers can't do anything that nthey're not explicitly programmed to. And Ar thur Samuel's checkers program was maybe nthe first I think really convi ncing refutation of this clai m. Namely, Arthur Samuel nmanaged to write a checkers program that could play checkers much better than he npersonally could, and this is an instance of maybe computers learning to do things that nthey were not programmed explicitly to do. nHere's a more recent, a more modern, more formal definition of machine learning due to nTom Mitchell, who says that a well-posed le arning problem is defined as follows: He nsays that a computer program is set to lear n from an experience E with respect to some ntask T and some performance measure P if its performance on T as measured by P nimproves with experience E. Okay. So not only is it a definition, it even rhymes. nSo, for example, in the case of checkers, th e experience E that a program has would be nthe experience of playing lots of games of checkers against itself, say. The task T is the ntask of playing checkers, a nd the performance measure P will be something like the nfraction of games it wins against a cert ain set of human opponents. And by this ndefinition, we'll say that Arthur Samuel's ch eckers program has learned to play checkers, nokay? nSo as an overview of what we're going to do in this class, this class is sort of organized ninto four major sections. We're gonna talk about four major topics in this class, the first nof which is supervised learning. So le t me give you an example of that. nSo suppose you collect a data set of housing prices. And one of the TAs, Dan Ramage, nactually collected a data set for me last week to use in the example later. But suppose that nyou go to collect statistics about how much hous es cost in a certain geographic area. And nDan, the TA, collected data from housing pr ices in Portland, Oregon. So what you can do nis let's say plot the square footage of the house against the list price of the house, right, so nyou collect data on a bunch of houses. And let' s say you get a data set like this with nhouses of different sizes that are li sted for different amounts of money. nNow, let's say that I'm trying to sell a hous e in the same area as Portland, Oregon as nwhere the data comes from. Let's say I have a hou se that's this size in square footage, and nI want an algorithm to tell me about how much should I expect

my house to sell for. So there are lots of ways to do this, and some of you may have seen elements of what I'm nabout to say before. nSo one thing you could do is look at this data and maybe put a straight line to it. And then nif this is my house, you may then look at the straight line and predict that my house is ngonna go for about that much money, right? There are other decisions that we can make, nwhich we'll talk about later, which is, well, what if I don' t wanna put a straight line? nMaybe I should put a quadratic function to it. Maybe that fits the data a little bit better. nYou notice if you do that, the price of my house goes up a bit, so that'd be nice. "'And this sort of learning problem of learning to predict housing prices is an example of nwhat's called a supervised learning problem. And the reason that it's called supervised nlearning is because we're providing the algorithm a data set of a bunch of square nfootages, a bunch of housing sizes, and as well as sort of the right answer of what the nactual prices of a number of houses were, right? nSo we call this supervised learning because we're supervising the algorithm or, in other nwords, we're giving the algorithm the, quote, right answer for a number of houses. And nthen we want the algorithm to learn the association between the inputs and the outputs nand to sort of give us more of the right answers, okay? nIt turns out this specific example that I drew here is an example of something called a nregression problem. And the term regression sort of refers to the fact that the variable nyou're trying to predict is a continuous value and price. nThere's another class of supervised learning problems which we'll talk about, which are nclassification problems. And so, in a classification problem, the variable you're trying to npredict is discreet rather than continuous. So as one specific example — so actually a nstandard data set you can download online [i naudible] that lots of machine learning npeople have played with. Let's say you collect a data set on breast cancer tumors, and you nwant to learn the algorithm to predict whether or not a certain tumor is malignant. nMalignant is the opposite of benign, right, so malignancy is a sort of harmful, bad tumor. nSo we collect some number of features, some number of properties of these tumors, and nfor the sake of sort of having a simple [inaudible] explanation, let's just say that we're ngoing to look at the size of the tumor and depending on the size of the tumor, we'll try to nfigure out whether or not the tumor is malignant or benign. nSo the tumor is either malignant or benign, and so the variable in the Y axis is either zero nor 1, and so your data set may look something like that, right? And that's 1 and that's nzero, okay? And so this is an example of a classification problem where the variable nyou're trying to predict is a discreet value. It 's either zero or 1. nAnd in fact, more generally, there will be many learning problems where we'll have more nthan one input variable, more than one input feature and use more than one variable to try nto predict, say, whether a tumor is malignant or benign. So, for example, continuing with nthis, you may instead have a data set that looks like this. I'm gonna part this data set in a nslightly different way now. And I'm making this data set look much cleaner than it really nis in reality for illustration, okay? nFor example, maybe the crosses indicate malignant tumors and the "O"s may indicate nbenign tumors. And so you may have a data set comprising patients of different ages and nwho have different tumor sizes and where a cross indicates a malignant tumor, and an n"O" indicates a benign tumor. And you may want an algorithm to learn to predict, given a nnew patient, whether their tumor is malignant or benign. '"So, for example, what a learning algorithm may do is maybe come in and decide that a nstraight line like that separates the two classes of tumors really well, and so if you have a nnew patient who's age and tumor size fall over there, then the algorithm may predict that nthe tumor is benign rather than malignant, okay? So this is just another example of nanother supervised learning problem and another classification problem. nAnd so it turns out that one of the issues we' ll talk about later in this class is in this nspecific example, we're going to try to predict whether a tumor is malignant or benign nbased on two features or based on two inputs, namely the age of the patient and the tumor nsize. It turns out that when you look at a real data set, you find that learning algorithms noften use other sets of features. In the breast cancer data example, you also use properties nof the tumors, like clump thickness, uniformity of cell size,

uniformity of cell shape, n[inaudible] adhesion and so on, so va rious other medical properties. nAnd one of the most interesting things we'll ta lk about later this quarter is what if your ndata doesn't lie in a two-dimensional or th ree-dimensional or sort of even a finite ndimensional space, but is it possible — what if your data actually lies in an infinite ndimensional space? Our plots here are two-dime nsional space. I can't plot you an infinite ndimensional space, right? And so it turns out that one of the most successful classes of nmachine learning algorithms — some may call support vector machines — actually takes ndata and maps data to an infinite dimensi onal space and then does classification using not ntwo features like I've done here, but an infinite number of features. nAnd that will actually be one of the most fa scinating things we talk about when we go ndeeply into classification al gorithms. And it's actually an in teresting question, right, so nthink about how do you even represent an in finite dimensional vector in computer nmemory? You don't have an infinite amount of computers. How do you even represent a npoint that lies in an infinite dimensional sp ace? We'll talk about that when we get to nsupport vector machines, okay? nSo let's see. So that was supervised learning. The second of the four major topics of this nclass will be learning theory. So I have a friend who teaches math at a different nuniversity, not at Stanford, and when you talk to him about his work and what he's really nout to do, this friend of mine will — he's a ma th professor, right? — this friend of mine nwill sort of get the look of wonder in his eyes, and he'll tell you about how in his nmathematical work, he feels like he's disc overing truth and beauty in the universe. And nhe says it in sort of a real ly touching, sincere way, and then he has this — you can see it nin his eyes — he has this deep appreciation of the truth and beauty in the universe as nrevealed to him by the math he does. nIn this class, I'm not gonna do any truth and beauty. In this class, I'm gonna talk about nlearning theory to try to convey to y ou an understanding of how and why learning nalgorithms work so that we can apply these lear ning algorithms as effectively as possible. nSo, for example, it turns out you can prove su rprisingly deep theorems on when you can nguarantee that a learning algorithm will wo rk, all right? So think about a learning ""algorithm for reading zip codes. When can you prove a theorem guaranteeing that a nlearning algorithm will be at least 99.9 per cent accurate on reading zip codes? This is nactually somewhat surprising. We actually prove theorems showing when you can expect nthat to hold. nWe'll also sort of delve into learning theo ry to try to understand what algorithms can napproximate different functions well and also try to understand things like how much ntraining data do you need? So how many exampl es of houses do I need in order for your nlearning algorithm to recognize the pattern be tween the square footage of a house and its nhousing price? And this will help us answer questions like if you're trying to design a nlearning algorithm, should you be spending more time collecting more data or is it a case nthat you already have enough data; it would be a waste of time to try to collect more. nOkay? nSo I think learning algorithms are a very powerful tool that as I walk around sort of nindustry in Silicon Valley or as I work with various businesses in CS and outside CS, I nfind that there's often a huge differen ce between how well so meone who really nunderstands this stuff can appl y a learning algorithm versus so meone who sort of gets it nbut sort of doesn't. nThe analogy I like to think of is imagine you were going to a carpentry school instead of na machine learning class, right? If you go to a carpentry school, they can give you the ntools of carpentry. They'll give you a hamme r, a bunch of nails, a screwdriver or nwhatever. But a master carpenter will be able to use those tools far better than most of us nin this room. I know a carpen ter can do things with a hammer and nail that I couldn't npossibly. And it's actually a littl e bit like that in machine learning, too. One thing that's nsadly not taught in many courses on machine l earning is how to take the tools of machine nlearning and really, really apply them well. nSo in the same way, so the tools of machin e learning are I wanna say quite a bit more nadvanced than the tools of carpentry. Maybe a carpenter will disagree . But a large part of nthis class will be just givi ng you the raw tools of machine learning, just the algorithms nand so on. But what I plan to do throughout this entire quarter, not just in the segment of nlearning theory, but

actually as a theme r unning through everything I do this quarter, will nbe to try to convey to you the skills to real ly take the learning al gorithm ideas and really nto get them to work on a problem. nIt's sort of hard for me to stand here and say how big a deal that is, but when I walk naround companies in Silicon Valley, it's co mpletely not uncommon for me to see nsomeone using some machine learning algorith m and then explain to me what they've nbeen doing for the last six months, and I go, oh, gee, it should have been obvious from nthe start that the last six months, you've been wa sting your time, right? nAnd so my goal in this class, running th rough the entire quarter , not just on learning ntheory, is actually not only to give you the tools of m achine learning, but to teach you nhow to use them well. And I've noticed this is something that really not many other "'classes teach. And this is something I'm rea lly convinced is a huge deal, and so by the nend of this class, I hope all of you will be master carpenters. I hope all of you will be nreally good at applying these learning algor ithms and getting them to work amazingly nwell in many problems. Okay? nLet's see. So [inaudible] the board. After lear ning theory, there's a nother class of learning nalgorithms that I then want to teach you a bout, and that's unsupervised learning. So you nrecall, right, a little ea rlier I drew an example like this , right, where you have a couple of nfeatures, a couple of input vari ables and sort of malignant tumors and benign tumors or nwhatever. And that was an example of a s upervised learning problem because the data nyou have gives you the right answer for each of your patients. The data tells you this npatient has a malignant tumor; this patient has a benign tumor. So it had the right nanswers, and you wanted the algorithm to just produce more of the same. nIn contrast, in an unsupervised learning problem , this is the sort of data you get, okay? nWhere speaking loosely, you're given a data se t, and I'm not gonna tell you what the right nanswer is on any of your data. I'm just gonna give you a data set and I'm gonna say, "Would you please find interesting structure in this data set?" So that's the unsupervised nlearning problem where you're sort of not given the right answer for everything. nSo, for example, an algorithm may find structure in the data in the form of the data being npartitioned into two clusters, or clustering is sort of one example of an unsupervised nlearning problem. nSo I hope you can see this. It turns out that th ese sort of unsupervised learning algorithms nare also used in many problems. This is a scr een shot — this is a picture I got from Sue nEmvee, who's a PhD student here, who is a pplying unsupervised learning algorithms to ntry to understand gene data, so is trying to look at genes as individuals and group them ninto clusters based on properties of what ge nes they respond to — based on properties of nhow the genes respond to different experiments. nAnother interesting application of [inaudible] sorts of clus tering algorithms is actually nimage processing, this which I got from Steve Gules, who's another PhD student. It turns nout what you can do is if you give this sort of data, say an image, to certain unsupervised nlearning algorithms, they will then learn to group pixels together and say, gee, this sort of npixel seems to belong together , and that sort of pixel seems to belong together. nAnd so the images you see on the bottom — I guess you can just barely see them on there n— so the images you see on the bottom are groupings — are what the algorithm has done nto group certain pixels together. On a small di splay, it might be easier to just look at the nimage on the right. The two images on the botto m are two sort of identical visualizations nof the same grouping of the pixe ls into [inaudible] regions. nAnd so it turns out that this sort of clustering algorithm or this sort of unsupervised nlearning algorithm, which learns to group pixels together, it turns out to be useful for nmany applications in vision, in co mputer vision image processing. '"I'll just show you one example, and this is a rather cool one that two students, Ashutosh nSaxena and Min Sun here did, wh ich is given an image like this, right? This is actually a npicture taken of the Stanford campus. You can apply that sort of cl ustering algorithm and ngroup the picture into regions. Let me actually blow that up so that you can see it more nclearly. Okay. So in the middle, you see the lines sort of groupi ng the image together, ngrouping the image into [inaudible] regions. nAnd what Ashutosh and Min did was they then applied the learning algorithm to say can nwe take this clustering and us e it to build a 3D

model of the world? And so using the nclustering, they then had a lear ning algorithm try to learn what the 3D structure of the nworld looks like so that they could come up with a 3D model that you can sort of fly nthrough, okay? Although many people used to th ink it's not possible to take a single nimage and build a 3D model, but using a lear ning algorithm and that sort of clustering nalgorithm is the first step. They were able to. nI'll just show you one more example. I like this because it's a picture of Stanford with our nbeautiful Stanford campus. So again, taking th e same sort of clustering algorithms, taking nthe same sort of unsupervised learning algor ithm, you can group the pixels into different nregions. And using that as a pre-processing step, they eventually built this sort of 3D model of Stanford campus in a single picture. You can sort of walk into the ceiling, look naround the campus. Okay? This actually turned out to be a mix of supervised and nunsupervised learning, but the unsupervised lear ning, this sort of cl ustering was the first nstep. nSo it turns out these sorts of unsupervised — clustering algorithms are actually routinely nused for many different problems, things like organizing computing clusters, social nnetwork analysis, market segmentation, so if you're a marketer and you want to divide your market into different segments or diffe rent groups of people to market to them nseparately; even for astronomical data an alysis and understanding how galaxies are nformed. These are just a sort of small sample of the applications of unsupervised learning nalgorithms and clustering algorithms that we 'll talk about later in this class. nJust one particularly cool example of an uns upervised learning algorithm that I want to ntell you about. And to motivate that, I'm gonna tell you about what's called the cocktail nparty problem, which is imagine that you're at some cocktail party a nd there are lots of npeople standing all over. And you know how it is, right, if you're at a large party, neveryone's talking, it can be sometimes very hard to hear even the person in front of you. nSo imagine a large cocktail party with lots of people. So the problem is, is that all of these people talking, can you separate out the voice of just the person you're interested in ntalking to with all this loud background noise? nSo I'll show you a specific example in a second, but here's a cocktail party that's I guess nrather sparsely attended by just two people. But what we're gonna do is we'll put two nmicrophones in the room, okay? And so becau se the microphones are just at slightly ndifferent distances to the two people, and th e two people may speak in slightly different nvolumes, each microphone will pick up an overl apping combination of these two people's ""voices, so slightly different overlapping voice s. So Speaker 1's voice may be more loud non Microphone 1, and Speaker 2's voice may be louder on Microphone 2, whatever. nBut the question is, given these microphone reco rdings, can you separate out the original nspeaker's voices? So I'm gonna play some audi o clips that were collected by Tai Yuan nLee at UCSD. I'm gonna actually play for you the original raw microphone recordings nfrom this cocktail party. So this is the Microphone 1: nMicrophone 1: nOne, two, three, four, five, six, seven, eight, nine, ten. nMicrophone 2: nUno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve, diez. nInstructor (Andrew Ng) : So it's a fascinating cocktail party with people counting from none to ten. This is the second microphone: nMicrophone 1: nOne, two, three, four, five, six, seven, eight, nine, ten. nMicrophone 2: nUno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve, diez. nInstructor (Andrew Ng) : Okay. So in supervised learning, we don't know what the nright answer is, right? So what we're goi ng to do is take exactly the two microphone nrecordings you just heard and give it to an unsupervised learning algorithm and tell the nalgorithm which of these discover structure in the data [inaudible] or what structure is nthere in this data? And we actually don't know what the right answer is offhand. nSo give this data to an unsupervised lear ning algorithm, and what the algorithm does in nthis case, it will discover that this data can actually be explained by two independent nspeakers speaking at the same time, and it can further separate out the two speakers for nyou. So here's Output 1 of the algorithm: nMicrophone 1: nOne, two, three, four, five, six, seven, eight, nine, ten. nInstructor (Andrew Ng) : And there's the second algorithm: nMicrophone 2: ""Uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve, diez. nInstructor (Andrew Ng): And so the algorithm discovers that, gee, the structure nunderlying the data is really th at there

are two sources of so und, and here they are. I'll nshow you one more example. This is a, well, th is is a second sort of different pair of nmicrophone recordings: nMicrophone 1: nOne, two, three, four, five, six, seven, eight, nine, ten. nMicrophone 2: n[Music playing.] Instructor (Andrew Ng): So the poor guy is not at a cocktail party. He's talking to his nradio. There's the second recording: nMicrophone 1: nOne, two, three, four, five, six, seven, eight, nine, ten. nMicrophone 2: n[Music playing.] nInstructor (Andrew Ng) : Right. And we get this data. It's the same unsupervised nlearning algorithm. The algorithm is actually called independent component analysis, and nlater in this quarter, you'll see why. And then output's the following: nMicrophone 1: nOne, two, three, four, five, six, seven, eight, nine, ten. nInstructor (Andrew Ng): And that's the second one: nMicrophone 2: n[Music playing.] nInstructor (Andrew Ng): Okay. So it turns out that be yond solving the cocktail party nalgorithm, this specific cla ss of unsupervised learning algor ithms are also applied to a nbunch of other problems, like in text proces sing or understanding f unctional grading and nmachine data, like the magneto-encephalogram would be an EEG data. We'll talk about nthat more when we go and describe ICA or independent component analysis algorithms, nwhich is what you just saw. "'And as an aside, this algorithm I just showed you, it seems like it must be a pretty ncomplicated algorithm, right, to take this overlapping audio streams and separate them nout. It sounds like a pretty complicated thi ng to do. So you're gonna ask how complicated nis it really to implement an algorithm like this? It turns out if you do it in MATLAB, you ncan do it in one line of code. nSo I got this from Samuel Wyse at Toront o, U of Toronto, and the example I showed you nactually used a more complicated ICA algorithm than this. But nonetheless, I guess this is nwhy for this class I'm going to ask you to do most of your programming in MATLAB and nOctave because if you try to implement the sa me algorithm in C or Java or something, I ncan tell you from personal, painful experien ce, you end up writing pages and pages of ncode rather than relatively few lines of code. I'll also mention that it did take researchers nmany, many years to come up with that one line of code, so this is not easy. nSo that was unsupervised learning, and then the last of the four major topics I wanna tell nyou about is reinforcement learning. And this refers to problems where you don't do one-nshot decision-making. So, for example, in the supervised learning cancer prediction nproblem, you have a patient come in, you predict that the cancer is malignant or benign. nAnd then based on your prediction, maybe the pa tient lives or dies, and then that's it, nright? So you make a decision and then there's a consequence. You either got it right or nwrong. In reinforcement learning problems, you are usually asked to make a sequence of ndecisions over time. nSo, for example, this is something that my students and I work on. If I give you the keys nto an autonomous helicopter — we actually ha ve this helicopter here at Stanford, — how ndo you write a program to make it fly, ri ght? You notice that if you make a wrong ndecision on a helicopter, the consequence of crashing it may not happen until much later. And in fact, usually you need to make a w hole sequence of bad decisions to crash a nhelicopter. But conversely, you al so need to make a whole sequence of good decisions in norder to fly a helic opter really well. nSo I'm gonna show you some fun videos of lear ning algorithms flying helicopters. This is na video of our helicopter at Stanford flying using a contro ller that was learned using a nreinforcement learning algorithm. So this wa s done on the Stanford football field, and nwe'll zoom out the camera in a second. You'll sort of see th e trees planted in the sky. So nmaybe this is one of the most difficult aer obatic maneuvers flown on any helicopter under ncomputer control. And this controller, which is very, very hard for a human to sit down nand write out, was learned using one of these reinforcement learning algorithms. nJust a word about that: The basic idea behi nd a reinforcement learning algorithm is this nidea of what's called a reward function. What we have to think about is imagine you're ntrying to train a dog. So every time y our dog does something good, you say, "Good dog," nand you reward the dog. Every time your dog does something bad, you go, "Bad dog," right? And hopefully, over time, your dog will lear n to do the right things to get more of nthe positive rewards, to get mo re of the "Good dogs" and to ge t fewer of the "Bad

dogs." ''So the way we teach a helicopter to fly or any of these robots is sort of the same thing. nEvery time the helicopter crashes, we go, "B ad helicopter," and every time it does the nright thing, we go, "Good helicopter, " and over time it learns how to control itself so as to nget more of these positive rewards. nSo reinforcement learning is — I think of it as a way for you to specify what you want ndone, so you have to specify what is a "good dog" and what is a "bad dog" behavior. And nthen it's up to the learning algorithm to figure out how to maximize the "good dog" nreward signals and minimize the "bad dog" punishments. nSo it turns out reinforcement learning is applie d to other problems in robotics. It's applied nto things in web crawling and so on. But it's just cool to show videos, so let me just show na bunch of them. This learning algorithm was actually implemented by our head TA, Zico, of programming a four-legged dog. I guess Sam Shriver in this class also worked non the project and Peter Renfrew and Mike and a few others. But I guess this really is a ngood dog/bad dog since it's a robot dog. nThe second video on the right, some of the st udents, I guess Peter, Zico, Tonca working non a robotic snake, again using learning algorith ms to teach a snake robot to climb over nobstacles. nBelow that, this is kind of a fun example. Ashutosh Saxena and Jeff Michaels used nlearning algorithms to teach a car how to drive at reasonably high speeds off roads navoiding obstacles. nAnd on the lower right, that's a robot program med by PhD student Eva Roshen to teach a nsort of somewhat strangely configured robot how to get on top of an obstacle, how to get nover an obstacle. Sorry. I know the video's kind of small. I hope you can sort of see it. nOkay? nSo I think all of these are robots that I thi nk are very difficult to hand-code a controller nfor by learning these sorts of l earning algorithms. You can in relatively short order get a nrobot to do often pretty amazing things. nOkay. So that was most of what I wanted to say today. Just a couple more last things, but nlet me just check what questions you have righ t now. So if there are no questions, I'll just nclose with two reminders, which are after class today or as you start to talk with other npeople in this class, I just encourage you again to start to form project partners, to try to nfind project partners to do your project with. And also, this is a good time to start forming nstudy groups, so either talk to your friends or post in the newsgroup, but we just nencourage you to try to star t to do both of those today, okay? Form study groups, and try nto find two other project partners. nSo thank you. I'm looking forward to teaching this class, and I'll see you in a couple of ndays. ''[End of Audio] nDuration: 69 minutes '