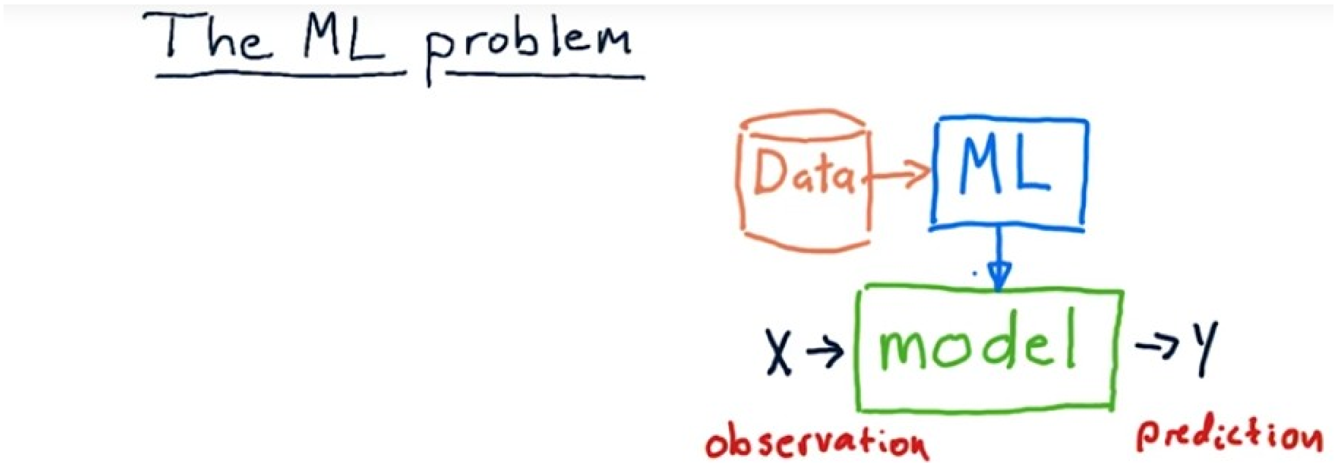
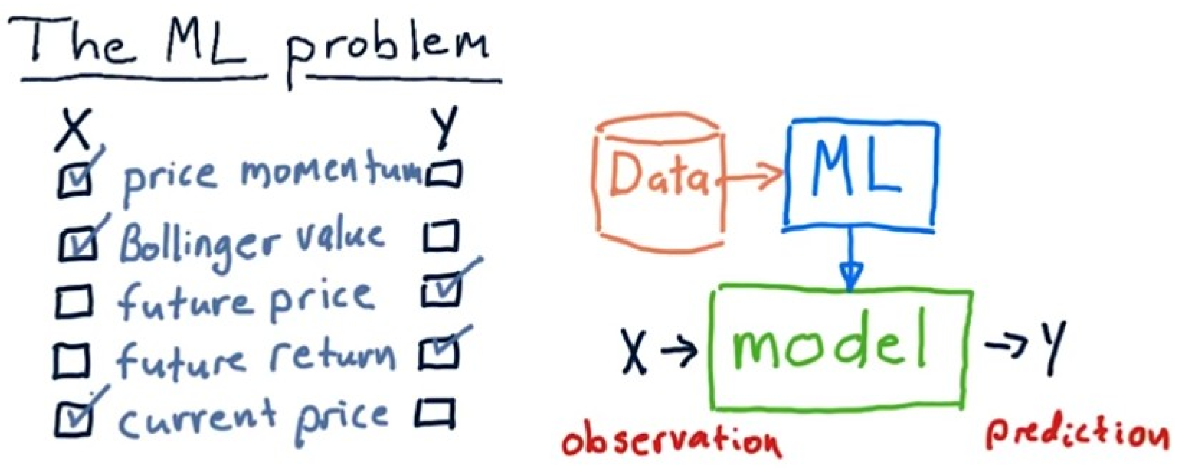
1. Welcome to the first lesson of machine learning algorithms for trading. In this lesson we're going to introduce how hedge funds and other financial institutions utilize machine learning. In general, the focus is on creating a model that can be used to predict future prices for stocks or other assets. Models like these have been around for a long time. What's different about machine learning is that it provides a suite of tools that support a data-centric way to build predictive models.

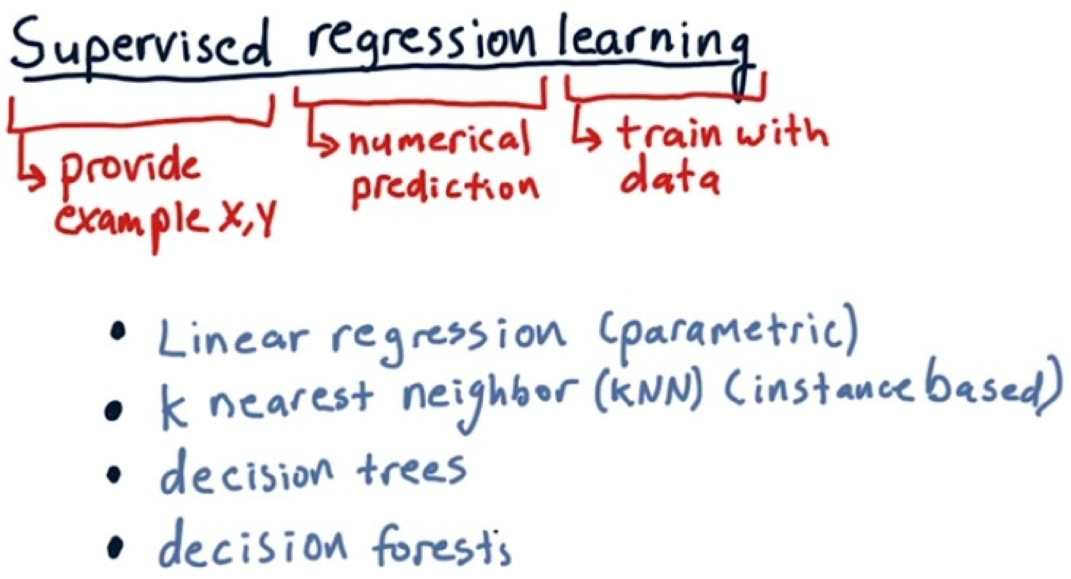


2. Scientists like to talk about the algorithms they build in terms of the problems that they solve, so this mini course is about machine learning and let's think about the problem the machine learning solves. In most cases, machine learning algorithms are focused on building a model. What's a model? A model is something that takes in observations like this x here, run it through some sort of process, and provide a y. So this y is typically a prediction, and the x is some sort of observation of the world. Examples might be, for us, x are some features of stocks and y is a future price. There are many other uses of machine learning models. That's just one of them. X can be multidimensional. In other words, there might be multiple factors that we're considering, might be Bollinger Bands, PE ratio, and so on. Y is typically single dimension and just represents that single-dimension prediction that we're trying to make. Now, there are lots of models that people have built that don't use machine learning at all. One example is the Black-Scholes model that predicts option prices. There are many other types of models that predict things that people build not using machine learning, but they develop mathematical formulas. But, of course, with machine learning, we're trying to use data. So the machine learning process is to take historical data, run it through a machine learning algorithm of some sort to generate the model. Then at runtime or when we need to use the model, we push x's in it and y's come out.

3. Now consider you were building a model and we were going to use it in trading in some way. Which factors might you consider as inputs or x's versus which of these might be appropriate outputs or y's for our model?



4. So the two things that make sense here as ys are future price, because we're trying to predict it, or future return, which is very similar. All these others are factors that we might use to predict these two. So price momentum is a predictive factor, Bollinger band value's a predictive factor, and current price might be as well.

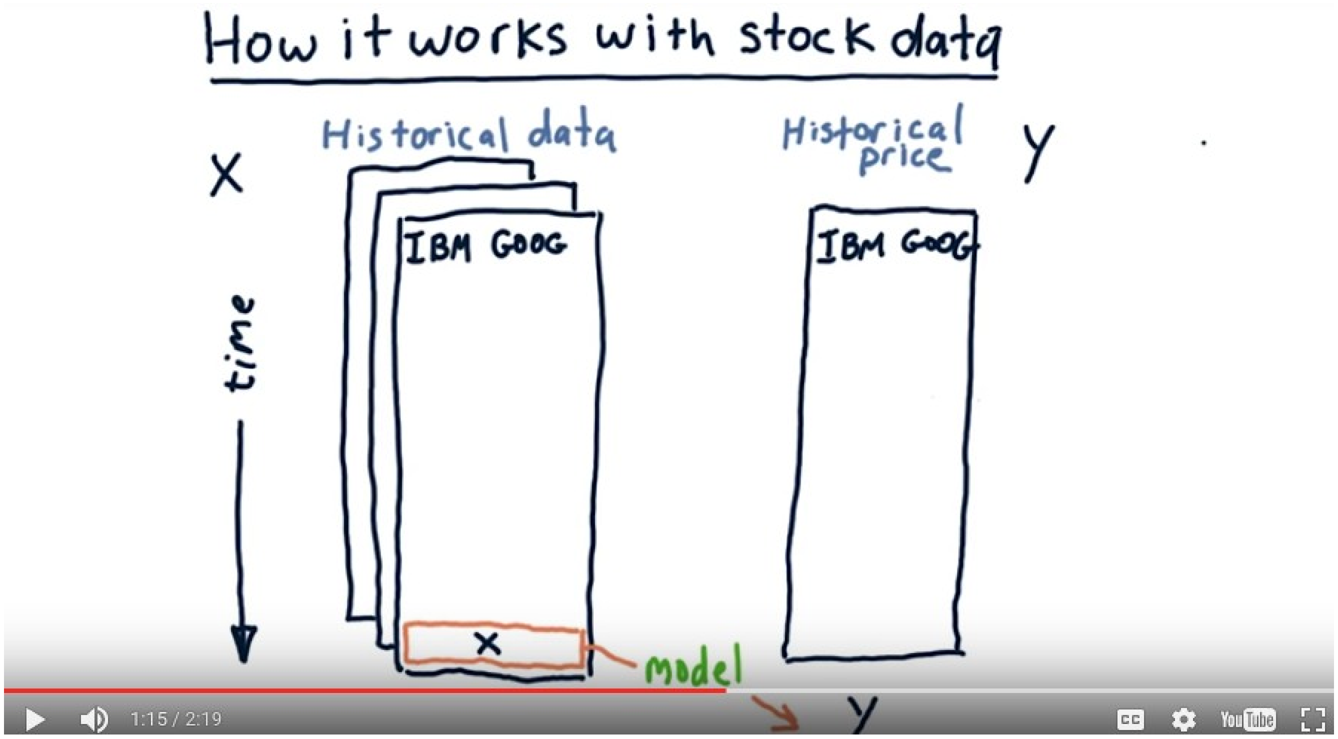

Supervised regression learning
provide example X,Y
numerical prediction
train with data
Linear regression (parametric)
k nearest neighbor (kNN) (instance based)
decision trees
decision forests

5. The particular flavor of machine learning that we're going to focus on in the first part of this course is called supervised regression learning. Those are a few big words. Let's break it down word by word. We'll start in the middle here with regression. Regression is a weird word. I don't think whoever chose it to represent what it means did a good job, but we're stuck with it. All that regression means is we're trying to make a numerical approximation or a numerical prediction. That's as opposed to classification learning where we might be trying to classify an object into one of several types as opposed to making a numerical prediction. Supervised means that we show the machine the x and also, if you will, the correct answer y. In fact, we show the machine many, many examples of x and y, and that's how it learns, okay, when I see this x, this is the y that's associated with it. That's the supervised component. Finally, when we say learning, what we mean is we are training with data. In this class we're taking historical stock data and training the system to make a prediction about the future, usually about price. There are a lot of algorithms that solve this problem and are supervised regression learning techniques. You've probably heard of linear regression and used it. Linear regression is a method that

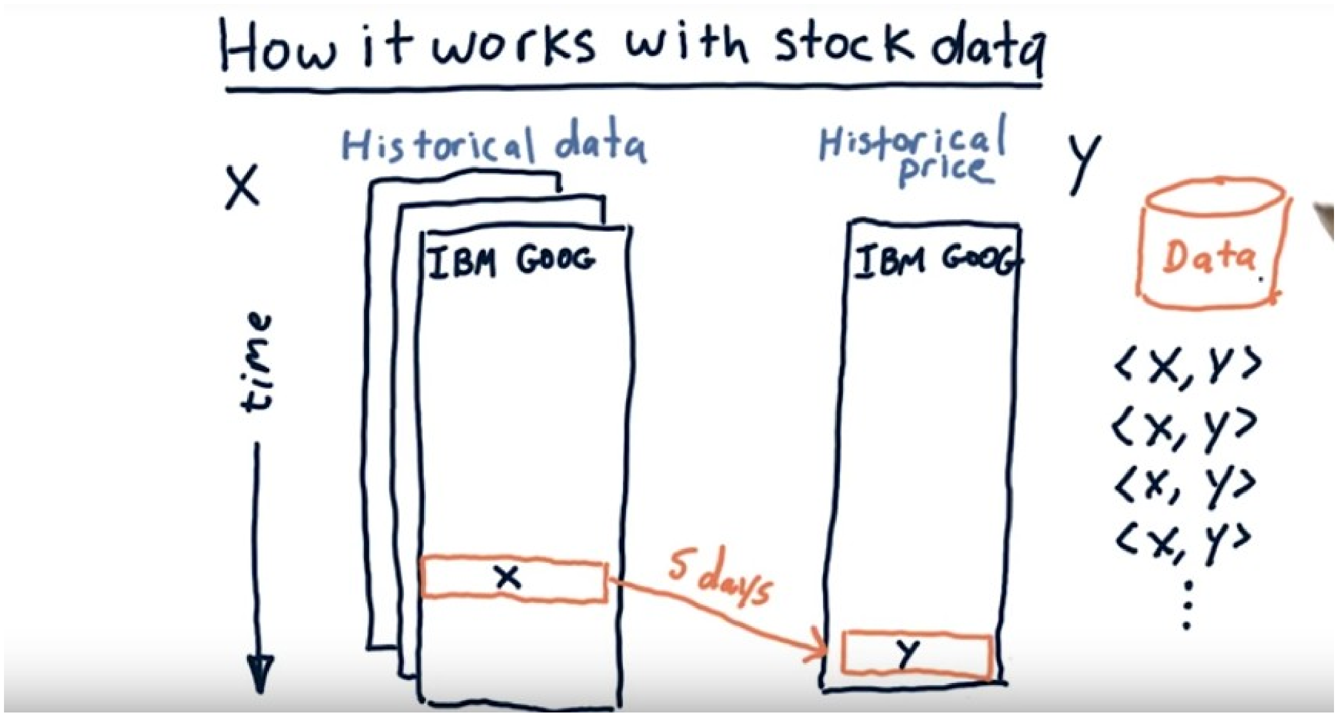
finds parameters for a model. So we call it parametric learning. K nearest neighbor, or KNN, is an extremely popular approach. And in fact, you're going to build a KNN learner as part of this class. What's different between parametric learning and instance based, which KNN is an example of, is that in parametric learning we take the data, munge it around to come up with a few parameters, and then throw the data away. In k nearest neighbor we keep all of this historic data, the x and y pairs, and when it's time to make a prediction we consult that data. That's what makes it instance based. Anyways, we'll spend a lot of time talking about k nearest neighbor. Two other techniques that are really popular are decision trees and decision forests. As you might guess, the way decision trees work is they store a tree structure and when a query comes in, it essentially bounces down that tree according to factors of the data. Each node in the tree represents essentially a question, is this x value greater than or less than this other value? And eventually we reach a leaf and that is the regression value that's returned. Decision forests are simply lots and lots of decision trees taken together, and you query each one to get an overall result. So this is the definition of supervised regression learning, which we're going to use a lot, and these are various algorithms that solve that problem.

6. I'd like to show you an example of supervised machine learning from my robotics past, this robot you see here is called Latgr or learning applied to ground robotics. It is using k nearest neighbor to learn how to navigate. Let me talk a little bit about what comes in and what goes out. It has a sensor in front of it that sees the world in front of it. It can see along a number of directions where there are obstacles or where the road is clear. So, the input is, in which directions are their obstacles. And also, it has the direction towards the goal. So in this case, the robots trying to get to a goal that's off the left, but these bushes that line the path are sort of preventing it from getting there. So x, in our case, are these perceptions. What do I see around me? And what's the direction to the goal? And y, or the output, Is, which direction to steer. So let's let it go for a while and see how well it can navigate based on a little bit of training that it's been given. It's doing pretty good here so far. But, it hasn't been

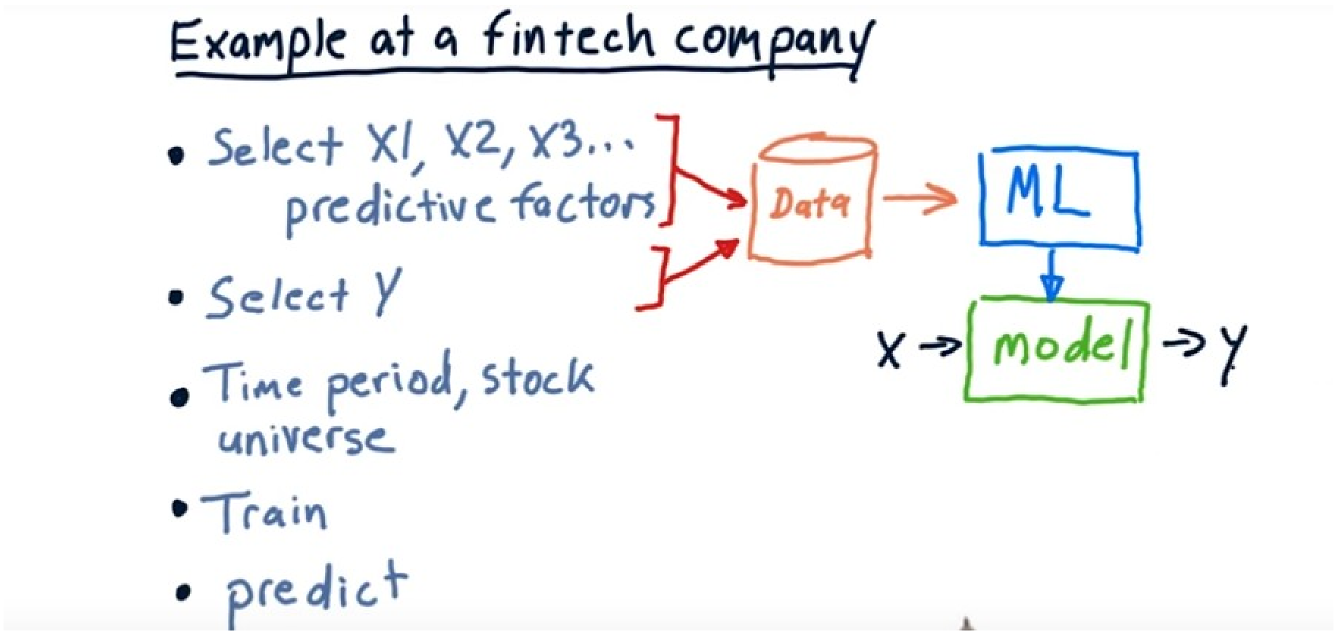
trained quite well enough and boom, it goes off the path. This is where learning comes in. We back it up over that area where it messed up, and we switch the learning switch on, and we're essentially tell it, okay, watch what we do, that's the correct thing to do in this case. So it takes all those examples of x's and y's, x's being the perception and y being the way it should respond, adds that to its memory, and now when it's running live, it consults this memory. It says oh, I see this, what should I do?. It finds the k nearest neighbors to that current observation and looks at what they say, and they essentially vote on what direction the robot should go. Now after we've trained it for about a half hour or so, it's able to navigate through difficult scenarios really quite easily. So keep in mind, this robot's driving autonomously, it's just consulting its memory for when I saw this, what was the right answer, why? Supervised learning, and it's able to get around quite well. Here's another more complicated scenario. The goal where it's trying to get is on the other side of that tree. So that's supervised learning for a robot and, again, the model maps these perceptions x to y, what it should do. And finance instead of Y being which way the robot should go, it's often a prediction about what a future price is. Okay, enough about robots, let's move on.



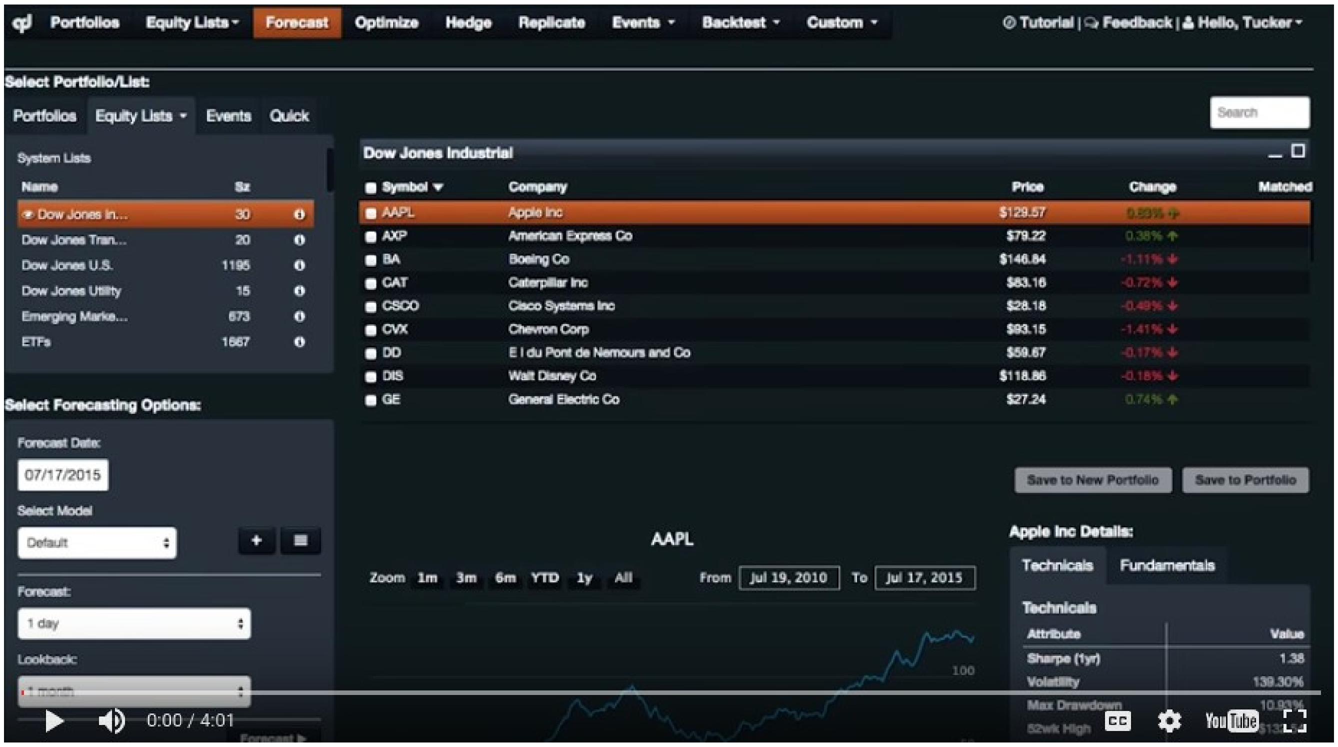
7. Let's consider how we can use this same approach now for stock data. This is one of our pandas Data Frames that contains some factors or features of stocks, and it's arranged in the usual way where each column represents the value of the feature for a particular stock. And time goes downward essentially. Now we might have many features for each stock, for instance we might have Bollinger Bands, momentum, PE ratio, and so on. We represent that by stacking these one behind the other. These are our X's. What is our Y? In most cases we want to use our historical feature data to predict a future price. But to train our model, we're going to use historical price as well. So the value of these factors or features today we call X and we like to be able to run that through the model that we've built and get Y which is our predicted future price. Well, we don't have that model yet, we've got to learn it. And we've got to learn it from data.



So here's how we do that. We roll back time so that we're back in history here at our first data point. We look at the values of our features there, and then we look, say, 5 days into the future to see what that future price is. So now we've got a pairing of these features with that future Y. We save that XY pair into our data. Now remember X can be multi-dimensional. And that's one instance of data. We move forward one day. So we've got a new set of X's and a new Y. And we record that in our database. Eventually, we reach a point where we can't go any further because there's no more Y data. In other words, if we went one day past this, Y would be out here in the future. So we don't have that data to use, and that means we've got some leftover data. In any case, we now, for each of these days in history, have a pair of X's and Y's that we can feed into our database to build our model.


Example at a fintech company
• Select X1, X2, X3... predictive factors
• Select Y
• Time period, stock universe
• Train
• predict
Data
ML
X
model
Y

8. I'd like to show you now how we use this process to build a machine-learning based forecaster at a FinTech company, at my company, Lucena Research. The first step is to select which factors do you want to use? So those are our Xs. Remember, it can be multi-dimensional. So these are things like Bollinger Bands, PE ratio and so on. Measurable quantities about a company that can be predictive of its stock price. The next step is to select what is it you want to predict? Usually we want to predict change in price, market relative change in price, or for now we could just think about that as future price. These both become our data, our Xs and our Ys that we used to train the model. Now that we know what our predictive factors are and what Y is going to be, we need to consider the breadth and depth of the data that we're going to use to train the system with. So that includes, for instance, time period. How far back in time do you want to go to train the system? And what's your stock universe? What universe of data, which symbols are you going to use to train the system as well? Now we can train our model. We can take this data to produce that model. We unleash our machine learning algorithm. This might be kNN. It might be a linear regression. It could be decision tree. Whatever. That algorithm takes this data and converts it into a model. We're ready now to use that model to do some prediction. The way we do that is we measure the quantities about the stocks that we want to make a prediction for now. We measure what those Xs are today, plug those into the model and the model should provide us our Y, or our prediction.



System Lists

Name	Sz
Dow Jones In...	30
Dow Jones Tran...	20
Dow Jones U.S.	1195
Dow Jones Utility	15
Emerging Marke...	673
ETFs	1667

Dow Jones Industrial

Symbol	Company	Price	Change	Matched
AAPL	Apple Inc	$129.57	0.53%	
AXP	American Express Co	$79.22	0.38%	
BA	Boeing Co	$146.84	-1.11%	
CAT	Caterpillar Inc	$83.16	-0.72%	
CSCO	Cisco Systems Inc	$28.18	-0.49%	
CVX	Chevron Corp	$93.15	-1.41%	
DD	E I du Pont de Nemours and Co	$59.67	-0.17%	
DIS	Walt Disney Co	$118.86	-0.18%	
GE	General Electric Co	$27.24	0.74%	

Select Forecasting Options:

Forecast Date: 07/17/2015
Select Model: Default
Forecast: 1 day
Lookback: 1 month

From Jul 19, 2010 To Jul 17, 2015

Apple Inc Details:

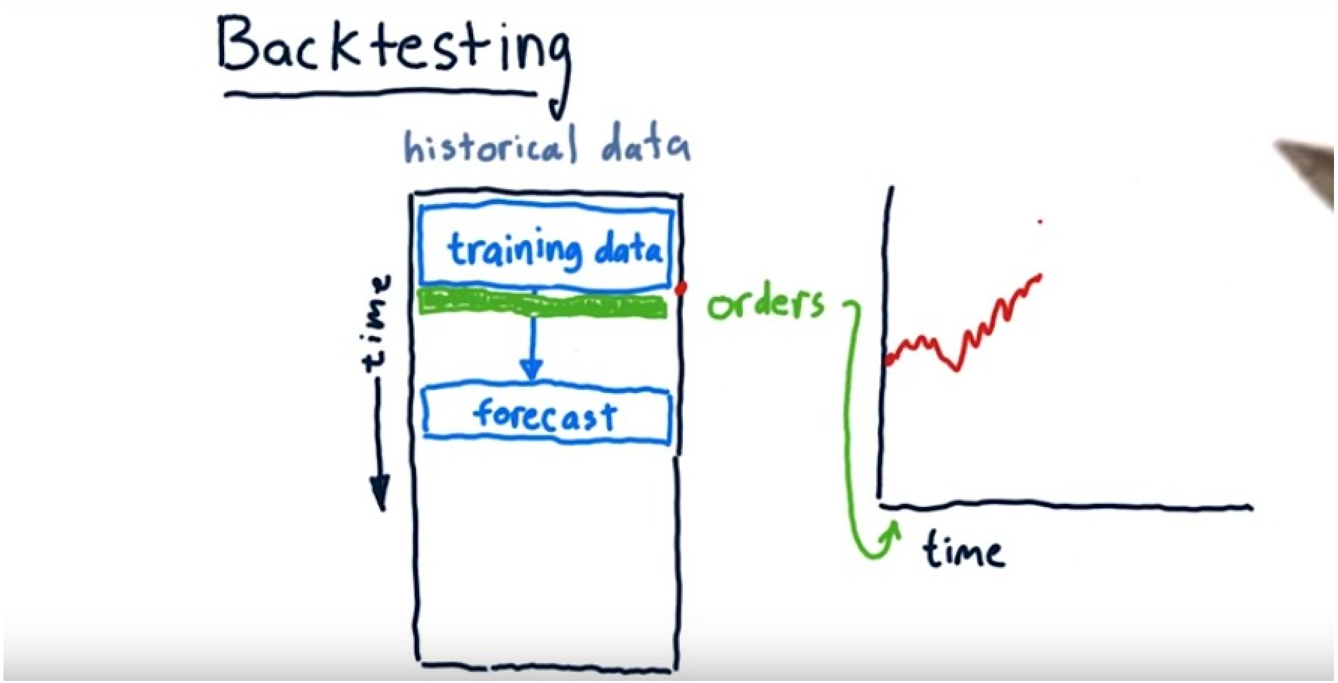
Technicals | Fundamentals

Technicals

Attribute	Value
Sharpe (1yr)	1.38
Volatility	139.30%
Max Drawdown	10.93%
52wk High	[illegible]

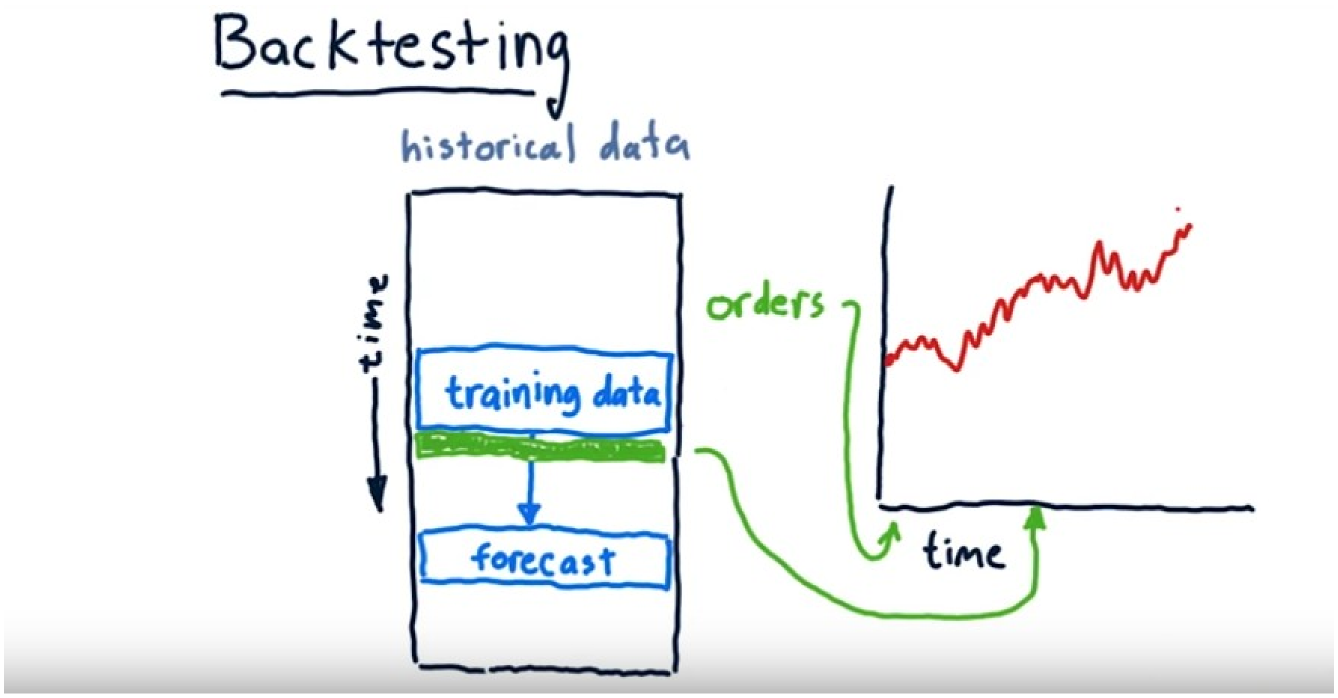
9. Now let me show you how we use these very same algorithms in a real application. This is a cloud-based application called QuantDesk. It was developed by a company I cofounded called Lucena Research. Now, the way it works is over on the left here, there are various lists of stocks that you can choose from. I've chosen Dow Jones, and when you click on that, you can see the list of stocks that make up that group of stocks. There are any number of ways that you can list particular stocks you're interested in looking at. Now, over here are our forecasting options, and this is how we tell the machine learning algorithm which factors we want to use, how far in the future do we want to predict, and so on. We're using right now the default model, and what that is is that's the list of factors that we think are important for making a future price prediction. So we can click here and see what those factors are. This is the list of factors that we're using now. So it turns out that these factors are

determined using another machine learning algorithm. We use a genetic algorithm for discovering these. That's a subject for a different lecture. But anyhow, these are the list of factors considered at present. Now, let's do a one week forecast. Let's make it a one month forecast. And let's use three months of data. So we're going to be looking back three months at all these factors. When we roll back time, we're able to see the future price of these stocks. So we can see how those factors presumably affected the future price. So let's do a forecast and see what the result looks like. So, this area here is the forecast for Apple. This is the historical price, looking back the last three months, and this is the forecast future price. So that line is indicating that we think it's going to go up, and this arc line above and below Is our confidence interval. We also report that data in a tabular format up here. So this is the current price, this is the forecast change in price, and the forecast percentage change. So, our system thinks that from today Apple is going to go up about 2.5%. Now, we also report some other information over here. We report what we call confidence and back tests score. Confidence refers to when we find those k nearest neighbors, how diverse are the ys that comes back. So, k is the number of neighbors, and when we look at all those neighbors, are the closest to the values of the quantities that Apple has today, we find the 30 closest ones, and we look at the standard deviation among all those 30. If they're very close, we're confident in our prediction. If they're spread apart, we're less confident. So you can see here, American Express is a more confident prediction, and, we rate this by a number of stars, by the way, where five stars represent our most confident estimates and one star, of course, our least confident. So our system thinks that American Express is going to go down by 0.9% over the next month. It's got a high confidence, and it's also got a high back test score. So what's this back test score? What we do is we roll back time, and we look over all this last three months and look forward one month. And we see how accurate all those predictions were over the last three months. The more accurate those were, the higher ranking we get there. So this is just an example of how we can use these same tools to make predictions in a live real system. And this is using, behind the scenes, the same software that you're learning in this class.

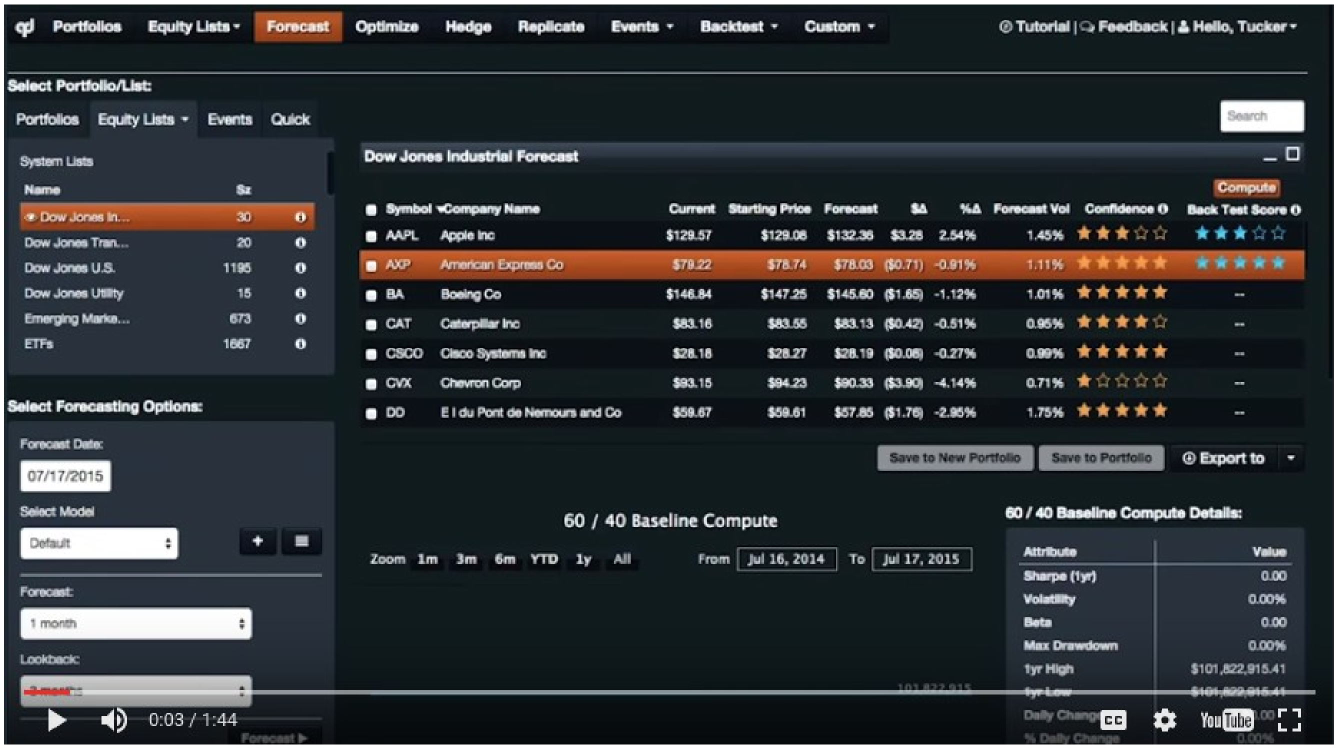


(本段沒講清楚, 可能也不重要)
10. Of course, a natural question is, well, how accurate are these forecasts? Can you act on them? Do they really predict the future? Well at least a first step towards answering that question can be found by

back testing. So that means you roll back time, and you test your system. So here's our historical data as usual it's organized with time coming down. Now in order to test the capability of our approach, we have to limit the data it sees to a certain amount of time and then make predictions into the simulated future. So we allow our system to only look at data up to a certain point. It can use data before that all it likes. It builds a model, and then makes a forecast. On the basis of that forecast, we can then now place orders, anticipating that that forecast will be achieved. So we might long some stocks or short them as appropriate. We can take those orders now, put them into our trading simulator and see how the portfolio works. So this time right here is the same as this time here, so we enter our positions on that date which was the same as that date and then we roll it forward and see what happens with the portfolio. Using software that we've built in this class, you can measure things like Sharpe ratio, return, and so on for that portfolio.



Now we're training over new data, make a new forecast and then make a new set of orders. We enter these orders in our trading simulator and then go forward and see what happens. I'm optimistic and it's always going to go up and to the right. Anyways, this process can be repeated over and over again using historical data and we can simulate our learning algorithm and how it would trade in this way. And that's called backtesting.



Select Portfolio/List:

Portfolios | Equity Lists | Events | Quick

System Lists

Name	Sz
Dow Jones In...	30
Dow Jones Tran...	20
Dow Jones U.S.	1195
Dow Jones Utility	15
Emerging Marke...	673
ETFs	1667

Select Forecasting Options:

Forecast Date: 07/17/2015

Select Model: Default

Forecast: 1 month

Lookback:

Dow Jones Industrial Forecast

Symbol	Company Name	Current	Starting Price	Forecast	$Δ	%Δ	Forecast Vol	Confidence	Back Test Score
AAPL	Apple Inc	$129.57	$129.08	$132.36	$3.28	2.54%	1.45%	★★★☆☆	★★★☆☆
AXP	American Express Co	$79.22	$78.74	$78.03	($0.71)	-0.91%	1.11%	★★★★★	★★★★★
BA	Boeing Co	$146.84	$147.25	$145.60	($1.65)	-1.12%	1.01%	★★★★★	--
CAT	Caterpillar Inc	$83.16	$83.55	$83.13	($0.42)	-0.51%	0.95%	★★★★☆	--
CSCO	Cisco Systems Inc	$28.18	$28.27	$28.19	($0.08)	-0.27%	0.99%	★★★★★	--
CVX	Chevron Corp	$93.15	$94.23	$90.33	($3.90)	-4.14%	0.71%	★☆☆☆☆	--
DD	E I du Pont de Nemours and Co	$59.67	$59.61	$57.85	($1.76)	-2.95%	1.75%	★★★★★	--

Save to New Portfolio | Save to Portfolio | Export to

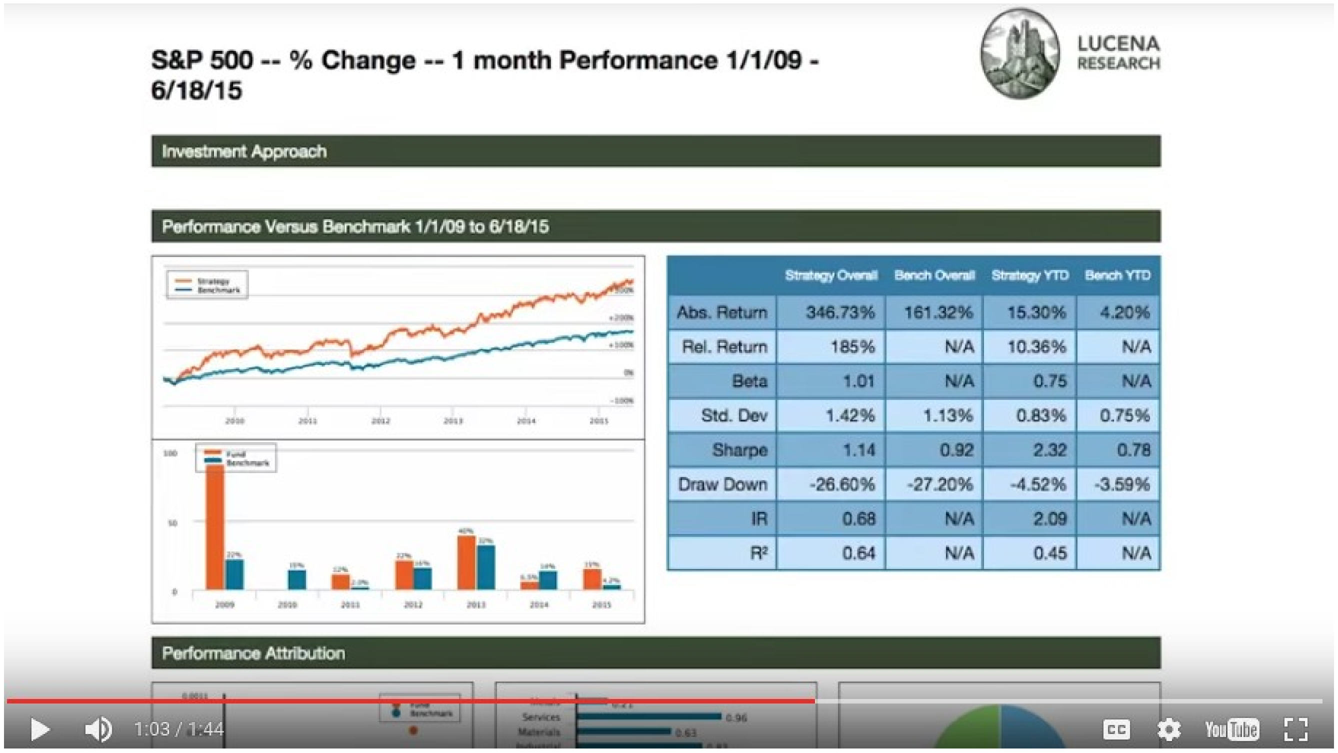
60 / 40 Baseline Compute

Zoom 1m 3m 6m YTD 1y All From Jul 16, 2014 To Jul 17, 2015

60 / 40 Baseline Compute Details:

Attribute	Value
Sharpe (1yr)	0.00
Volatility	0.00%
Beta	0.00
Max Drawdown	0.00%
1yr High	$101,822,915.41
1yr Low	$101,822,915.41
Daily Change	[illegible]
% Daily Change	0.00%

11. On the QuantDesk platform, we can backtest in exactly the same way. Go over here to the BackTest tab and click on Forecast. And again, we can configure which group of stocks are we going to work with, and what are the settings for our backtest. For instance, how much money do we start with? How frequently do we want to trade, over what period of time do we want to run the backtest and so on. Anyway, we make those selections. We also need to choose parameters for the forecaster. Again, which model do we want to use, which includes what are the factors that we're paying attention to. How far into the future do we want to make our forecast and how much data looking back do we want to use. Once we make all those selections, we can click on BackTest and it'll start running. It takes quite a bit of time, so we're going to go ahead and process in the background. And I'll show you what one of these backtests looks like.



S&P 500 -- % Change -- 1 month Performance 1/1/09 - 6/18/15

LUCENA RESEARCH

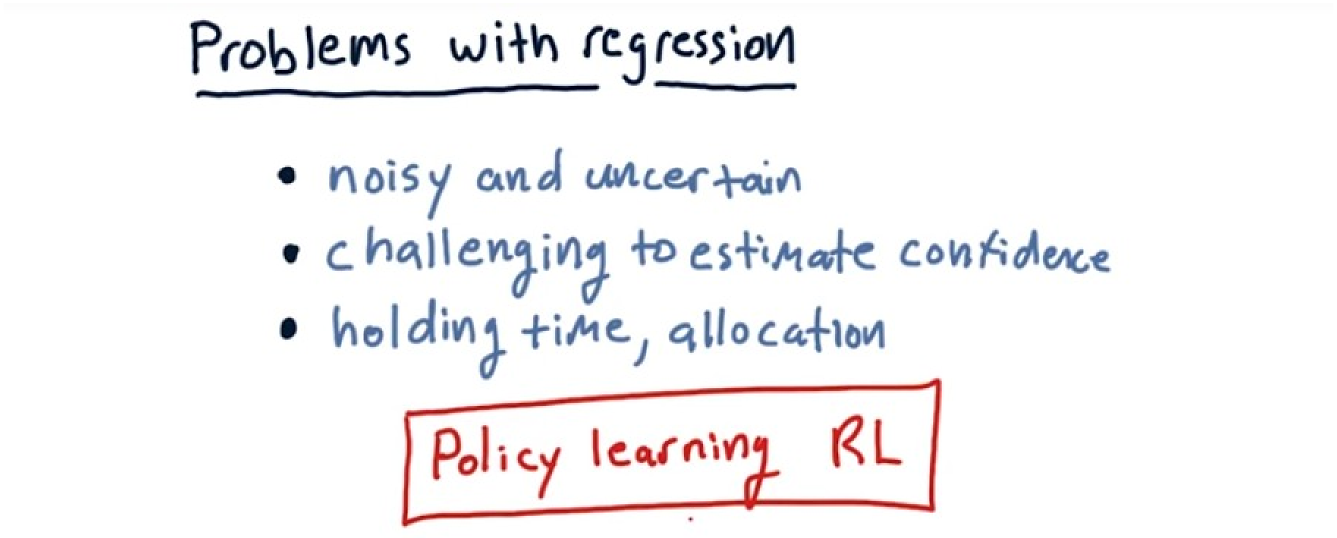
Investment Approach

Performance Versus Benchmark 1/1/09 to 6/18/15

	Strategy Overall	Bench Overall	Strategy YTD	Bench YTD
Abs. Return	346.73%	161.32%	15.30%	4.20%
Rel. Return	185%	N/A	10.36%	N/A
Beta	1.01	N/A	0.75	N/A
Std. Dev	1.42%	1.13%	0.83%	0.75%
Sharpe	1.14	0.92	2.32	0.78
Draw Down	-26.60%	-27.20%	-4.52%	-3.59%
IR	0.68	N/A	2.09	N/A
R²	0.64	N/A	0.45	N/A

Performance Attribution

Here's an example report their system generates. This is again for a forecast backtest. The orange line here represents the historical value of our portfolio. The blue is the benchmark that we're using, which is S&P 500. You can see here a number of metrics tabulated over this period of time we saw 346% return. Our sharp ratio was 1.14 and so on. So this is just an example of the kinds of systems you can build using the sorts of things we teach you in this class. So that's the forecast backtest.



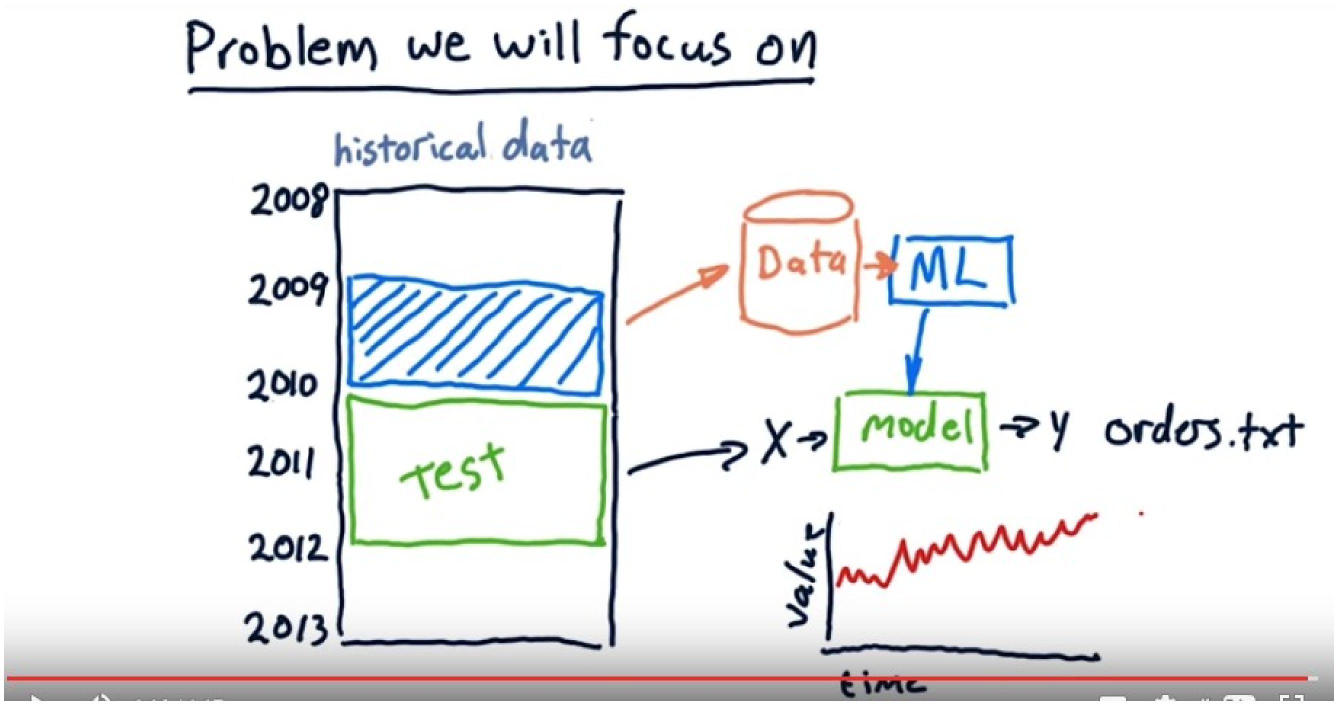
Problems with regression

- noisy and uncertain
- challenging to estimate confidence
- holding time, allocation

Policy learning RL

12. As you saw by that Beck test, regression-based forecasting can be useful. It's also worth noting that that particular Beck test was not spectacular. It did indeed beat the S&P 500, which is important, but it didn't beat it spectacularly. [LAUGH] Usually we find that performance in the real world Is not as awesome as in back testing. So we would probably still see good return from that strategy but it wouldn't be quite the same as we saw in that back test. So I want a list a few problems we see

sometimes with regression based forecasting. First of all, our forecasts always seem to be noisy and uncertain. So there is value in there, but it has to be accumulated over many trading opportunities. It's hard to know how confident you should be in a forecast. I mentioned in my software demo a moment ago that we could look at standard deviation of the nearest neighbors. That works okay. It's really not too strong of a measure, however. So it's difficult to know how confident you ought to be in any particular forecast. It would be nice if you could know because that would enable you to essentially bet less on forecasts that are less certain. Additionally, it's not clear how long you should hold a position that might have arisen from a forecast, and how you should allocate to that position. These are all challenges using regression based forecasting. Some of these issues can be addressed using reinforcement learning. Where instead of making a forecast of a future price, we had the system learn a policy and the policy tells the system whether to buy or sell a stock. We'll get to this towards the end of this class but it's an interesting alternative to the regression based approaches we're going to use here at the beginning of the class.



13. I'm going to take a moment here now to talk about the problem we're going to focus on for the rest of this class. We're going to look at a certain period of data, train our models over that period, and then make forecasts and trade over some other period. So here's our historical data. We're going to use the period of 2009 as our data to train our model. You'll be implementing several machine learning algorithms to create different models and will be comparing them one against another. Then we'll test over the years 2010 and So those testing values will become our X which will push through the model to create a Y or a forecast. And using this forecast you'll generate an orders.txt file which you can push through your market simulator. We'll see how that strategy performs, measure it's sharp ratio and its total return and so on. And that way we'll be able to compare different machine learning algorithms that generate these orders. That's it for this lesson. Our introduction to machine learning and how it works at hedge funds and other financial institutions. I will see you online again soon. Bye-bye.