## `[CAP4611-21Spring](https://webcourses.ucf.edu/courses/1369384/calendar_events/2158980)

# Variant 1

# Day 2 (Thurs, Jan 14):

## Supervised Learning

Supervised Learning techniques automatically learn a model of the relationship between a set of **descriptive features** and a **target feature** based on a set of historical examples, or **instances**.

Examples:

* Suitable for a loan? (Risk Assessment)
* Price Prediction
* Dosage Prediction
* Classification

**How does Machine Learning Work?**

It searches through a set of possible prediction models for the model that best captures the relationship between the descriptive features and a target feature in a dataset. We can look for models that are consistent with existing data but that can pose problems.

* First, with large datasets, there is likely noise, e.g., feature values mislabeled, in the data. Prediction models that are consistent for noisy data often make incorrect prediction models.
* Second, the training set only represents a small sample of possible sets of instances in the domain. Thus, ML is often referred to as an ill-posed problem, that is, a problem for which a unique solution cannot be determined using only the information available.

**Inductive Bias**

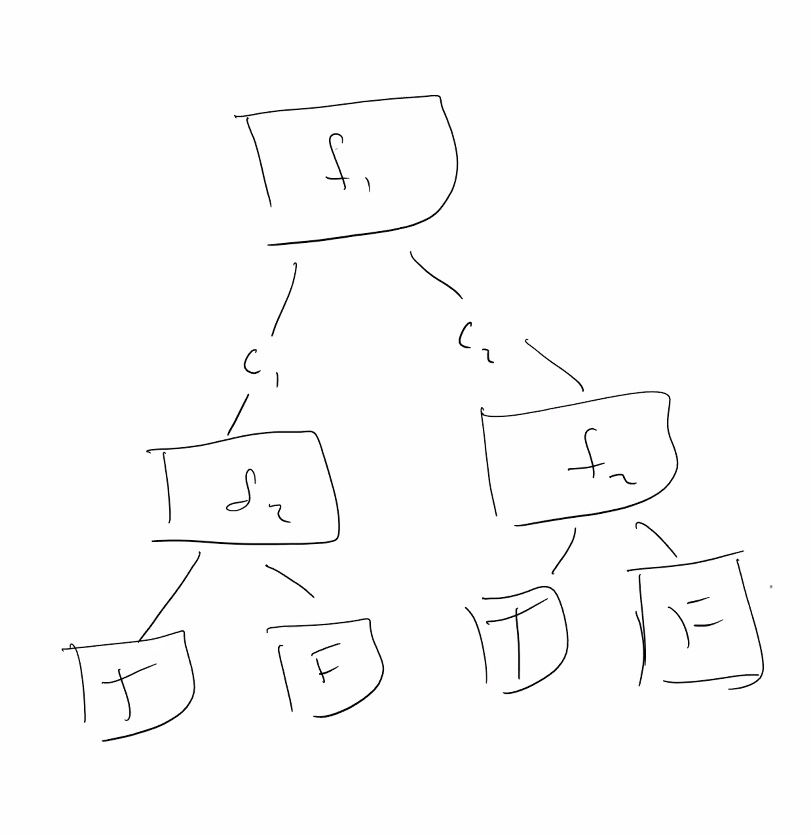
A way to restrict your models to models of a certain form. A set of assumptions that defines the model selection criteria of a machine learning algorithm.

Example:

Linear Regression is restricted to models that follow the form

Example 2:

A decision tree is restricted to models that follows the form



If a prediction model is to be useful, it has to be able to make predictions for queries not present in the dataset. A model that predicts these queries captures the underlying relationship between the descriptive and target features and is said to **generalize** well.

**Underfitting**

When the prediction model selected is too simplistic to represent the underlying relationship in the dataset between descriptive features and the target feature.

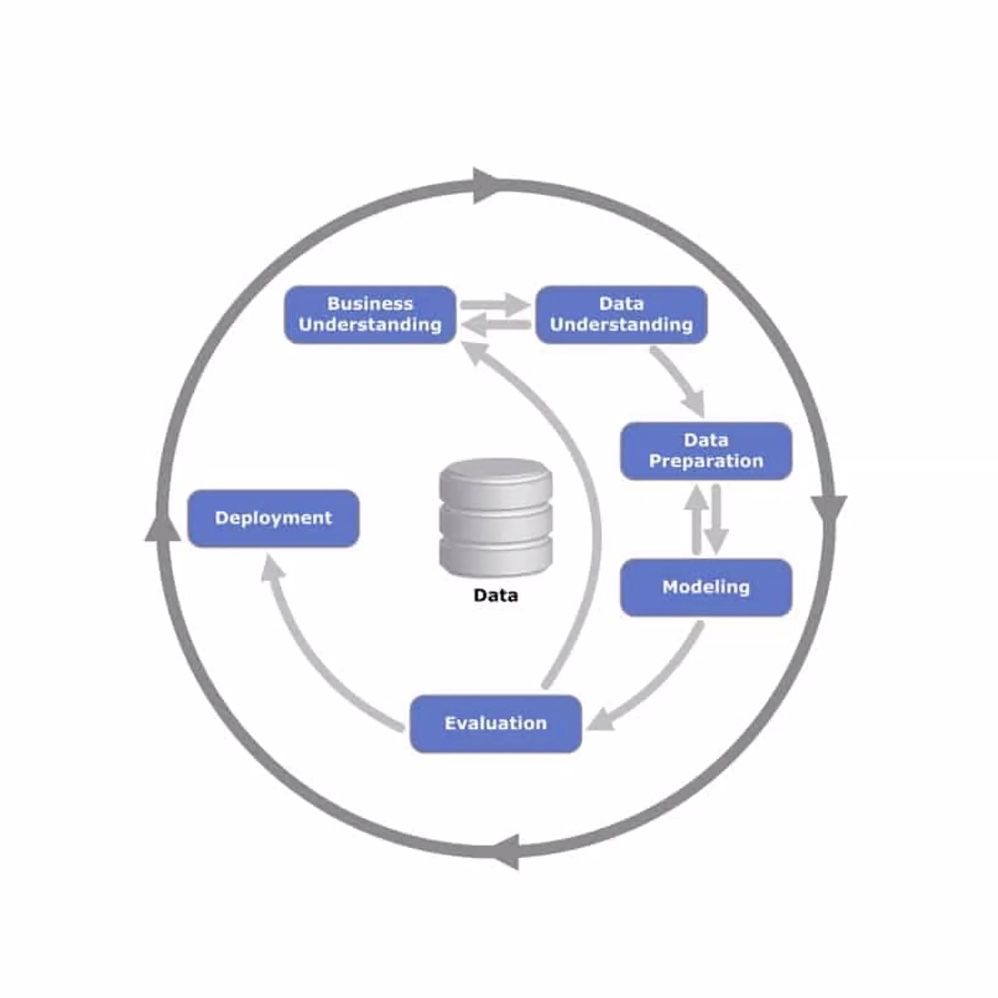
**Overfitting**

When the prediction model selected is so complex that the model fits the dataset too closely and becomes sensitive to noise in the data.

**Goldilocks Model**

It’s *just right*, between underfitting and overfitting.

# CRISP-DM Process



* **Business Understanding**
  + The first, primary, goal is to fully understand the business problem that is being addressed and to design the solution.
* **Data Understanding**
  + Data Collation: organizing data according to some set of rules. (Usually numerical ID value).
* **Data Preparation**
  + A majority of your time will be spent cleaning up the data.
  + Building models requires specific kinds of data organized as an **Analytics Base Table (ABT)**.
* **Modeling**
  + ML techniques are applied to produce predictive models. Models are then tweaked and refined, to make sure they sufficiently strike a balance between generalizing for new samples and being consistent with the current data.
* **Evaluation**
  + Determine whether or not the answers are correct. Need to show that accurate predictions are being made after the solution is deployed. (No underfitting or overfitting).
* **Deployment**
  + Integrate the machine learning model into the process within an organization

# Variant 2

Recap: gave intro last time, kaggle tutorials, stuff like that

Students should go through:

* Kaggle courses
* Github Tutorials
* Python tutorials (Numpy, Pandas, Sci-Kit, i think there was another)

You don’t need to use anaconda, use Kaggle, make your stuff public, this is so you don’t need to go through the pain of setting up an Anaconda environment.

This course was created last semester, this is the second time it is being taught. My understanding of why this course has come about is because at some point, the Ai course went away from drawed Ai and became too focused on ML. So a couple years ago, the ai course takes care of all the traditional stuff. However, now its more about Neural networks.

Berkeley has good courses on AI if you’re interested.

There is a reason that schools like Berkeley, Stanford, MIT etc etc. their quality of instruction is better than what you’ll find at random universities.

We won’t be working solely on ML but also some neural network stuff (supervised and unsupervised).

…

…

…

Talk about parler, tech companies, pirate bay,

…

…

…

“Are there any questions about basic admin stuff?”

Kaggle: we are gonna learn that,

Ethics: you should be ethical, you should think about those things.

What is Computer Ethics? Don’t be an A hole

More talk about R next Tuesday

What do we need to know about SQL:

SELECT()

Machine learning can be done in matlab, if you don’t want to use Python, but we’re not gonna do that.

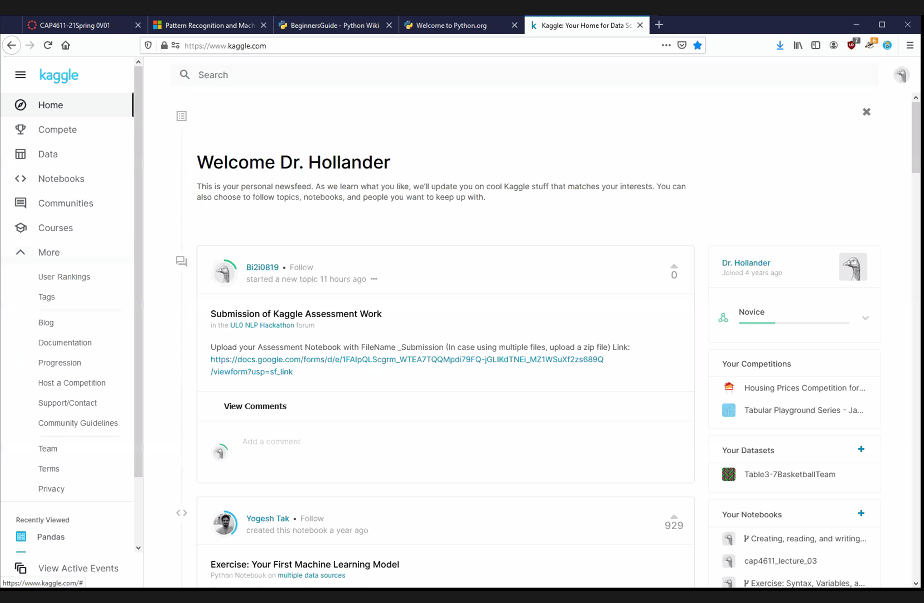
Did you know: there are people who graduate with CS degrees and can’t solve fizzbuzz

Syllabus quiz this week

If you don’t know Python, there are a bunch of tutorials all over the internet, and on webcourses.

**Recording begins:**

If you haven’t yet started to play with python:

* Make a kaggle account
* Login
* You’ll see something like this
* 
* Go to notebooks, you’ll see a bunch of publicly created notebooks
* The New Notebook button will allow you to write code on a web based interface
* This is a jupyter notebook
* Python is written in two ways
* A .py file, and write it using a script
* Running the Python command line (?) and typin in your code in that

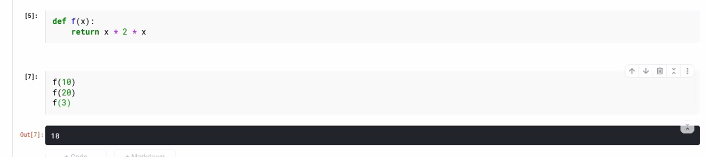
Repl.it:

* Everything you put inside one of their documents, treat it as a script. You click run and everything runs

Jupyter notebooks:

* Real fancy (?)
* Basically, a jupyter notebook is made of a bunch of cells. Variables, functions, and a bunch of other things when declared in the cells are global and are shared between instances.
* You can add datasets from your computer, OR import from online resource
* Examples include:
  + Covid-19 info
  + Smartphone
* The benefit with cells is that you don’t have to execute everything, rather you can execute one or two cells at a time.
* **Cells are not scoped independently (they’re global)**

Benefits of a notebook:

* A notebook cell will automatically output the last statement in it.
* 
* If you run a Jupyter notebook by clicking the run button **at the top** the code gets executed sequentially from top to bottom
* We can also add HTML(?)

For our assignments:

* We will be creating these kaggle notebooks
* Notebooks also have support for LaTeX, which allows you to write equations
* To submit assignments, just provide a link to your kaggle notebook
* For this class, set your notebook to public when you want to do these assignments
* Side note: Can’t use MS office to construct equations and migrate them into a kaggle notebook
* CBorne (?) will be used to generate plots and graphs
* If I require for you to write equations, it’ll be in LateX
* So if we look at the preview of the next lecture notes found in a kaggle notebook
* Exhibit A
* We will be using Pandas, and various other libraries to help us with
* The programming language R will not be something we will be heavily coverin.
* Pandas will be what we will be using to analyze our data
* We won’t start off with creating a ML algorithm, we gotta extract features, analyze the data, etc etc
* **Do not assume or depend on the instructor posting notes**

**Lecture 02:**

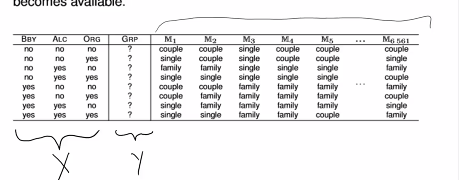
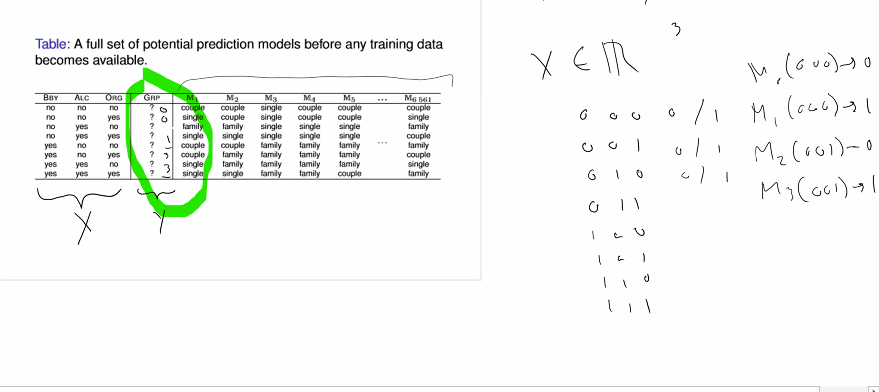
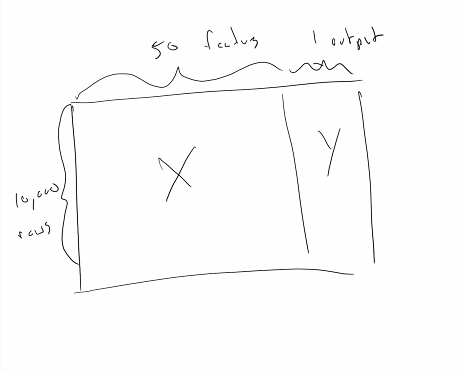
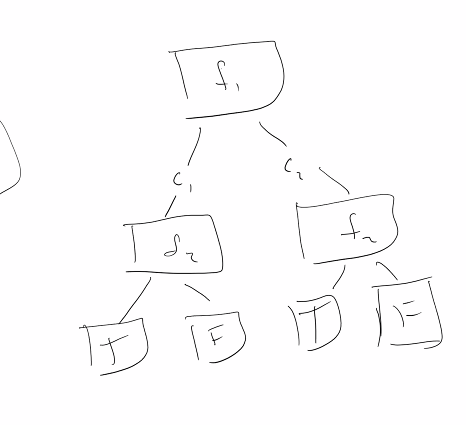
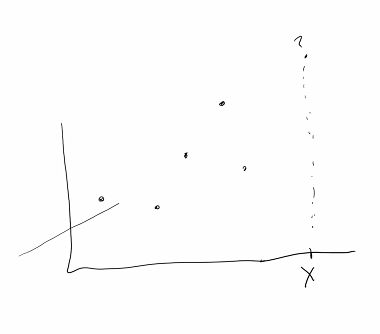
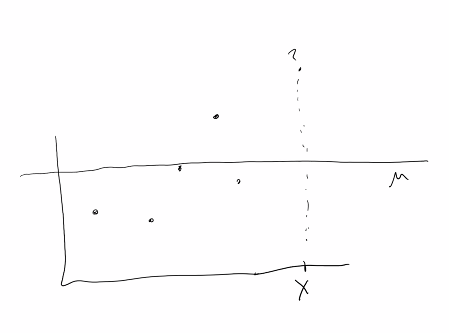
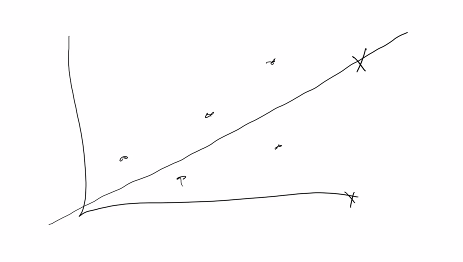
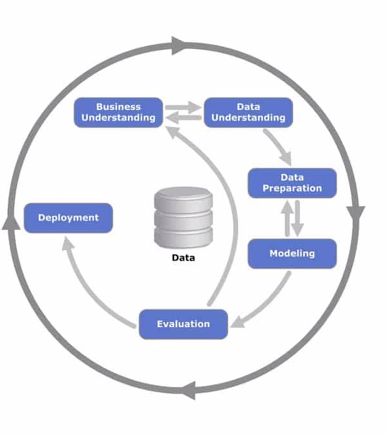
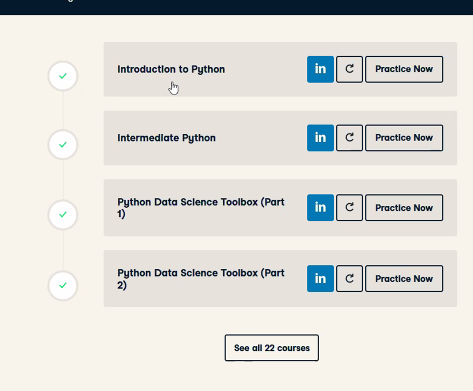
Supervised learning:

* (?)...

Unsupervised learning:

* Massive table of input data, is there a way to compress the data down and decrease the amount of featres

Example:

* We have a dataset
* 
* It has 3 inputs, some features(?)
* Before we do anything with ML:
  + We need to find statistical information about the table such as averages, outliers, etc
* If we think about the data
* The first thing we know is we have a bunch of inputs, but don’t know what corresponds to the output
* Before we do anything, we have a large number of models that might work on the data
* Each of these models will make a different output based on the input
* Let’s say Y is binary, X is binary:
  + X was 10 rows and 3 columns
  + Don’t worry about rows, look at columns (?)
  + How rows of input can we have?
  + We can have 16 models, where we might have
  + Model 0:
    - Takes in M1(0, 0, 0) ---> 0
    - Takes in M2(0, 0, 0) ---> 1
    - Takes in M3(0, 0, 0) ---> 0
    - Takes in M4(0, 0, 0) ---> 1
  + What happens when we know that certain groups are 0 or 1?
    - 
    - Well we can get rid of some of the models
    - The models that we have left are “consistent models”
    - Eventually as we continue to train,
    - What happens if we have a lot more features?
    - If we have 100 or 1000 features, what is the probability that we can get down to a single model
    - Very smol
    - Note: “not every model has to be a good model”
    - The main thing we want to get across is
    - How machine learning works, we have a large dataset
    - The learning process to reduce the amount of models (?)
    - So in an ideal world:
      * You will have enough data to just get a single model to describe every possible input(?) that you can have
      * But this never happens
    - Can you go from an infinite number of models to one?
      * Not realistically, think about counting all the numbers between 0 and 1 (uncountable infinity? Discrete II)
    - Unless you’re dealing with a large dataset, models will be inconsistent
  + Let’s say we have some data. It has 10,000 rows.
  + Are those 10,000 rows all possible combinations of data that can be imputed?
  + Probably not
  + So that dataset that you do have, think as it as a training set.   
      
      
    
  + The dataset represents a sample of a much larger population
  + The reason that this topic has been so big is because the datasets are progressively getting bigger and bigger (thx google)
  + Amazon has millions of people to analyze data. That isn’t the entire population…
  + So we take the input and train on it. We will get n models
  + The n models:
    - Will fit the data we have, not always the data we lack
  + **How well will these n models work on the stuff we have yet to see?**
  + Example
    - If you are in the dataset for predicting what movies you’ll watch, and you don’t change we can 100% predict what you’ll watch/linke
  + What we need is a generalization of the data.
    - This will be discussed later
  + Generalization is generalization
  + Abstraction is sort of treating every img of a dog instead of a specific kind of dog
  + When we are handling our models, we can set certain levels of abstraction on the data
  + If we chose to label the images of cats and dogs as cats and dogs, the machine will learn cat vs dog
  + However, if we enter the more specific features of the dogs and cats, we can get more specific output
  + How specific we want our network to be, all depends on the question and the answer we want
  + We are going to write and see the algorithms. We won’t see (?)
  + We’ll see regression algorithms, scalable vector machines,
  + We won’t need to derive our own algorithms.
* Assignments
  + We won’t need to code algorithms for scratch (performance issues)
  + It is more important as a first approach is to understand how to use the algorithms and what they output
  + Once we apply the algorithms, we gotta figure out which ones are best and doing certain things
  + By the end of the course, you should be able to do ML competitions
* Back to models
  + Machine learning works cus we reduce the number of models
  + The models themselves depend on what you can get working
* Inductive Bias
  + A way of restricting models to models of a certain model :L
  + If you think about regression:
    - y= a0x0 + a1x1 + a2x2 …. + anxn
  + **There is an inductive bias, as we are only considering models of this structure**
  + Lets say we have a feature, and it splits off into a couple of conditions:
    - 
  + A inductive bias in a decision tree biases the result by biasing to only consider decision trees
  + **What we need to do is run a bunch of different inductive biases**
  + Depending on which ones best, we deploy that one to production
  + SO ML uses inductive bias to reduce the number of biases until you can classify the number of biases
* The no free lunch theorem
  + In the absence of trained models, every model is great (?)
* The general practice method is to throw the kitchen sink at it
  + Some models will work well, others will not
* Note
  + Your gonna click run on ur program and ur gonna question whether it works or not.
    - Did it freeze, is it broken?
  + 2hrs later
    - Idk whats goin on
  + 8hrs later
    - ☺
  + 24 hrs later
    - It finishes
* Some ML algorithms will take days to train
* Google
  + Created a Trillion input (?) Model and got it to work the other day
* Question
  + Q: ML model has like a trillion parameters, were some of those blank or empty?
  + A: it was a massive amount of text that they were taking as input. A ton of data. Not every weight is used at the same time, but they do have values.
  + Think about it as x1, x2 … x1,000,000,000,000 (? trillion)
* TensorFlow
  + We’ll get to that during NN (neural networks)
* Regarding inductive bases
  + So let's say we have a data set that consist of two columns of data and because there are two columns of data we could represent that data as a set of points in two dimensional points
  + We would like to know where upon some line X is going to be
  + 
  + How many different lines can we draw through these 5 points?
  + A whole bunch.
  + Which line can we draw that will give us a good prediction of the data?
  + (My guess is linear regression)
  + We could use this line
  + 
  + Is that a good prediction?
    - It works
  + But its not the best prediction
  + The mean of a dataset is a default order prediction(?)
  + The mean is the baseline model
  + The mean is kinda wrong about everything
  + In this situation we have something called **underfitting**
* Underfitting:
  + **The data is underpredicting everything, it is underfitting.**
* The other situation that we might have, is that we have the same dataset, want to predict value of x, we could draw somethin like:
* 
* How incorrect is this line against the dataset?
  + It predicts all the known values within the dataset, but won’t be a good predictor of the new value of x
  + **This is called overfitting**
* Now, if you have a dataset for which you have every possible combination of inputs:
  + Overfitting is great! Congrats (this is not realistic)
  + This would be considered a lookup table then
* In the real world we would want:
  + A line that generalizes like this:
  + 
  + This line will produce some bad predictions
  + But it will give us data that is way better than anything that underfits without overfitting the data.
  + (something like a basis of a vector space in linear algebra)
* Neural networks
  + A bunch of functions composed together, nonlinear
* Is some data nongeneralizable?
  + No.
  + It depends on how it is sampled
* Example
  + Survey
    - If you only select a small group of people from a very specific demographic
    - Cannot generalize, the data is inherently biased, a sample bias
* Sample Bias
  + When data isn’t representative of the larger population
  + We cannot learn anything useful, we can learn about the small demographic
* Solving machine learning problems
* CRSIP-DM (Late 90s)
  + “Got abandoned”
  + Still the widely used algorithm for solving machine learning problems
* Example of CRISP-DM
* 
* Similar to the software engineering cycle
  + If we don’t know what the problem is, any ML algorithm is useless
* Example
  + ML offers a solution where we can detect people doing fraud
  + However the bank actually wants to know how to **prevent** fraud
* We want to understand what the data means
  + Once we do this, we can go to data preparation
* Data preparation
  + Edit the data, update values, normalize data, manipulate etc etc
  + **Most of the time in ML is here**
  + In the real world:
    - Somebody wants you to analyze support logs for automation
    - Once you look at the text transcripts and all of the data,
    - Inaccuracies within the data due to:
      * Typos
      * Euphemisms
      * Incorrect Data
      * Etc etc
    - There are courses on cleaning up data
  + “Clean Data” refers to the process of getting rid of the errors above
* Modeling phase
  + We build the model,
* Evaluation
  + Are the answers generated by the model correct?
  + If no, go back to modeling, **or worse, back to data preparation**
  + Once we are happy, we can move on to deployment
* Deployment
  + Not covered in this course
  + Involves integrating the model into something useful (backend stuff probalbly)
* A really big thing that happens is:
  + When your doing facial recognition…
    - The datasets do not have a racial representation reflective of the real world
  + Example:
    - Was doing projects for some class
    - Somegroup wanted to make a robot that shoot a nerf gun
    - “The nerf gun shot an african american in the front row” when it was not supposed to (?)
    - Problems like this are being recognized and are slowly being posted
* Next week
  + Exploratory data analysis,
  + data analogy,
* First assignment:
  + Sometime next week after we learn Kaggle, K-Nearest-Neighbor (KNN), etc
* **Coding competitions:**
  + **Are a way for corporations to get cheap work**
  + **Competitions could be used to solve hard problems that none of the employees know how to solve and do not want to pay for some expert**
* Data.gov:
  + Thats the US government website for all things data. It has a crap load of data that staticicians use a lot
* QUIZZES are due:
  + Idk, maybe last part of class, or sometime during the week
* In order to prep for next week class:
  + Aside from the two tutorials that I linked,
  + Datacamp: you need an account
    - Use your knights.ucf.edu account
    - Once you have access you have it for 6 months
    - Look into “Data scientist for Python” and do the first few courses:
    - 
    - Not formal assignments, as you should just want to do it yourself
  + Mechanical Turk
    - Crowdsource labelling
    - Labelling is agony
  + Quiz next week:
    - About Data analysis material (?)
* Questions:
  + How to prepare for quiz?
    - Show up to class
    - Should be about knowledge stuff, take 15 minutes to do
  + The official textbook:
    - Should be useful
  + Software to learn:
    - Python, Pandas, (??)
    - All the tutorials can be found on Datacamp
  + 2For this week on friday
    - Syllabus quiz due
  + Next week
    - Exploratory data analysis
  + The thing about python:
    - Really easy just to look it up on stackoverflow.com
    - Or anywhere on the internet
  + Quizzes will be:
    - Open material, **do not go to other people for help**

End of lecture

You can ping Dr. Holland on Discord