Before class:

…

Extra credit opportunity:

* Every 10 datacamp courses, you will get a flat 1% bump to your grade.
* **Final grades will not be rounded**
* **79.9999999999999 = C**
* Start with python stuff

Tip:

* Don’t focus solely on the Machine Learning aspects, you should also work on the data gathering process (Crap in ---> F(x) -----> Crap out)

Kaggle:

* There are competitions on kaggle, if you want to participate in that

…

Recap:

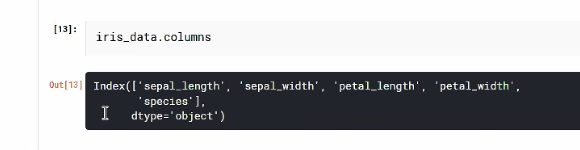
Exploratory Data Analysis:

* Examining the data to figure out the quality and various other aspects of it before passing it to the machine.
* Check to see all the columns you want are still there
* Once its loaded, figure out any problem areas
* We looked at ways to display information about statistics, qualitative data, and quantitative data
* As nice as numbers are, its sometimes better to see a picture, we’ve talked about the seaborn library to show graphs

Continuous values:

* We are concerned about how they spread out
* How they (?)

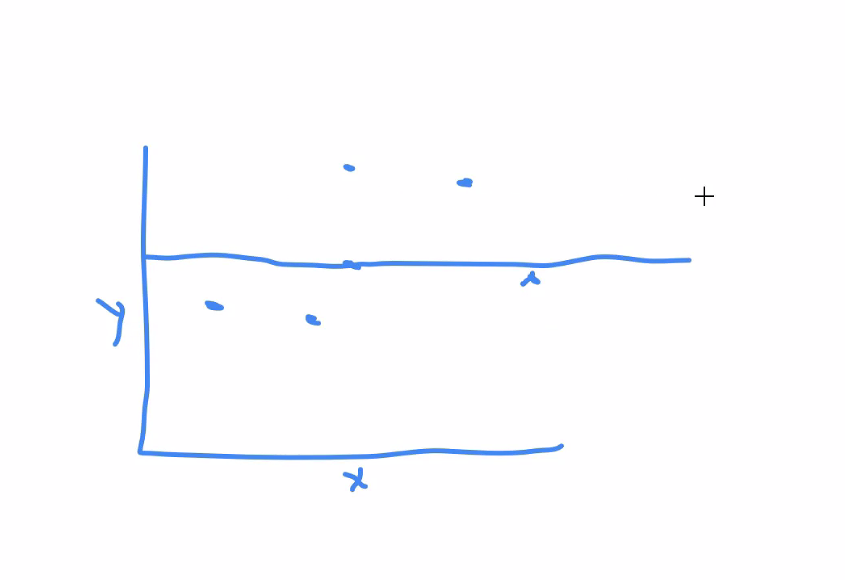
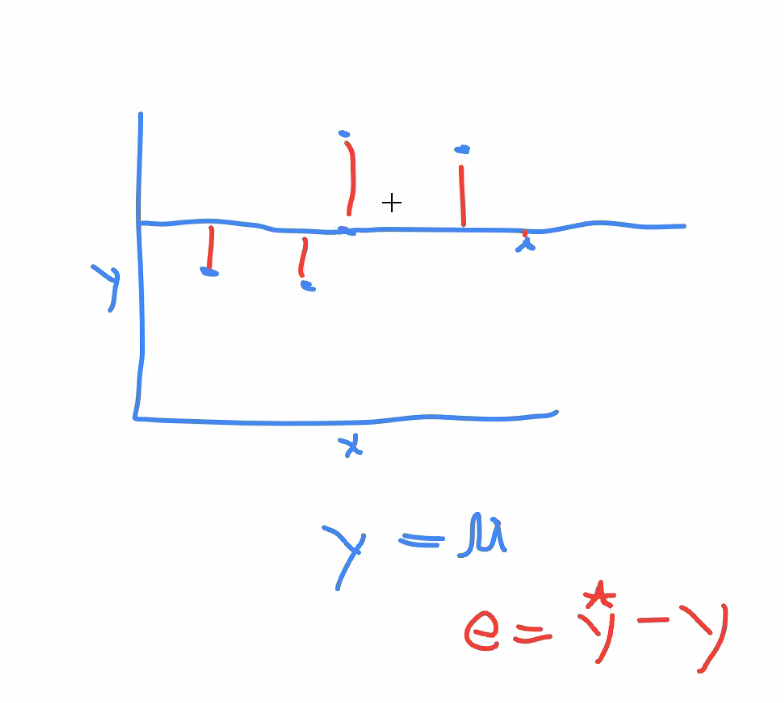
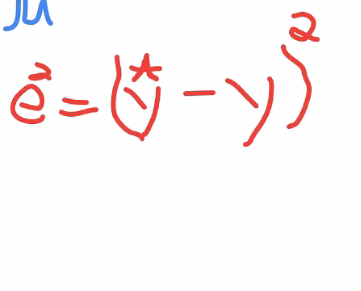
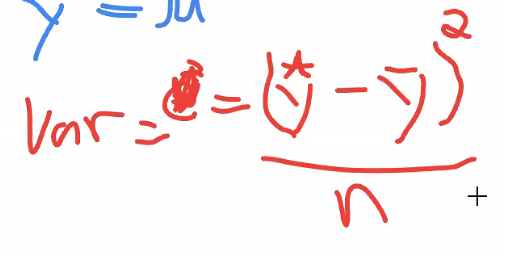
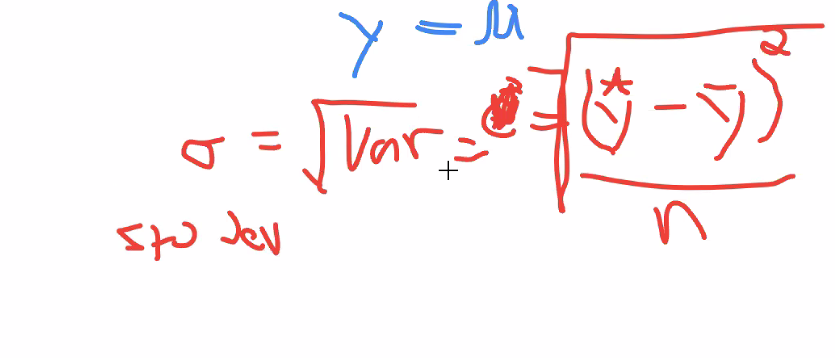
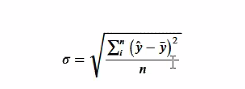
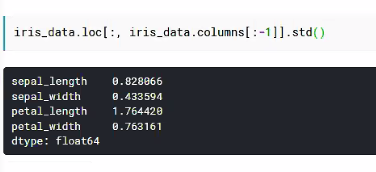
Example:

* Get the mean
* Iris\_data.loc[:, iris\_data.columns[:-1]].mean()
* Get the columns
* 
* We can use negative indices to look at the columns backwards (-1 access the last element)
* When accessing the last column through [:-1] it assumes the last column is the last column of the target vector (?)
* (?)

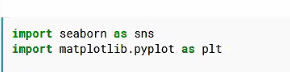
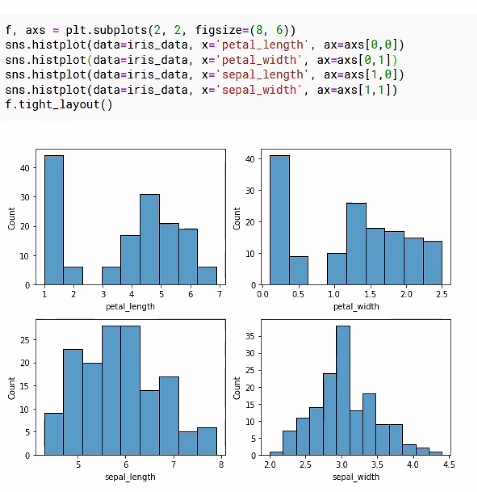
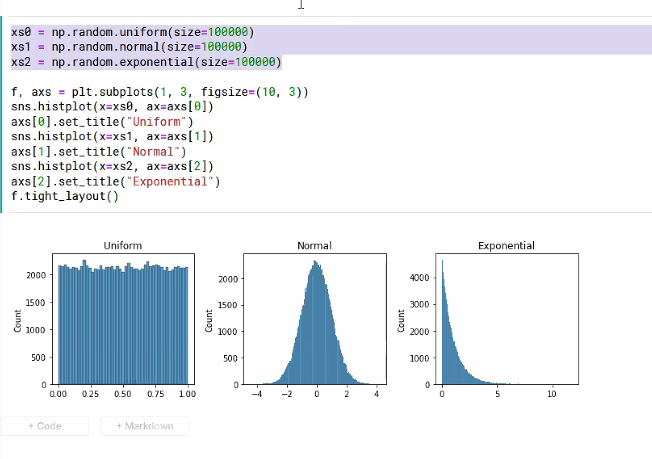
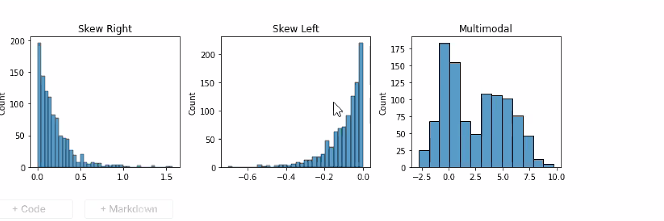
Mean

* If you don’t explicitly say what columns to evaluate, it may give unexpected results

Standard Deviation Recap:

* So you can imagine that if we had an x and y axis, and we had a bunch of points on this block
* So the mean is the average of all the values divided by the number of values
* The mean is a line that goes straight through the data
* 
* The error of the points is the distance between the points and that line
* y hat is the predicted value:
* 
* The difference between yHat and y is the error.
* The problem with the [red] formula is that negative distance doesn’t make sense.
* We can square the value to also exaggerate the error. So:
* 
* How do we find the average error?
* We just divide the total error by the number of points (?)
* This is also known as the variance, its the average square error
* 
* The problem with variance is the square units. The square units are not directly related to the mean.
* So people decided to create something called [sigma]. The square root of the variance:
* 
* And [Sigma] is the standard deviation.
* LaTeX:
* 
* Top equation should contain a square
* 
* 
* When you are talking to people, you prefer standard deviation because the units are simple
* In a computer you shouldn’t care

What’s a better way to look at a distribution?

* Lets actually look at the distribution:
* We can use a mixture of mathplotlib and pandas to generate what we’re looking for
* **Note the library taken from matplotlib**
* 
* 
* What sticks out looking at the petals and sepals?
* Petals appear to be bimodal
* Based on the data, can we differentiate between the sepals and the petals and predict what kind of flower it is?
* If we want to classify an iris into its species based on some measurement would we have better luck using the petals or the speals?
* Looking at the histogram, petals, there appear to be two groups
* Thats something useful to know.
* Now in general, there are many many different types of sample distributions, you can go to wikipedia and seee many
* Three most common ones are teh
* Uniform - pretty level
* Exponential - looks like x^n | n > 1
* Normal - bell curve
* 
* The code above generates a list of data that forms into one of the three shapes, and displays it
* A distribution can be skewed left or right
* 
* If the data is heavily skewed, is that caused by outliers that we need to exclude? Should we investigate further?
* When talking about skewness, we are talking about looking at the distribution rather than the actual data
* ...
* Mathematically, a uniform distribution has a (variance) of zero, depending on how you normalize it [?]
* What is the best choice of distribution for modeling my data?

Some of the things that you can think about when looking at a machine learning problem is that you’re looking for:

* Given that I have dataset X and a target vector Y…
* Given that I have an input vector X, and a target output Y, what is the probability that Y = ? given that I have X
* p(Y | X)
* What is the probability that X = ? given that Y is ??, p (X | Y)
* (We will elaborate on this later when talking about probability)

What are some things that can go wrong with data?

* Missing Values - We can have different measurements that are missing for each entry
* Irregular counts (cardinality) - in the case of regular counts, that is when you expect 5 categories, but the data has 43,000
* Outliers - Unexpected values within the dataset
* Incorrect inputs - Misspellings, euphemisms, etc etc
* If we have a row/column that is causing too many problems, we can remove it.
* The issue with this is that it can lead to bias

Example

* You have a dataset with 10 rows,
* 3 are A
* 3 are B
* 2 are C
* 1 is D
* Row with D has missing values.
* If we remove the D row, the algorithm will have no idea how to process D

Question:

Based on all this, lets say we have data similar to that, and we have all this data from a previous model, lets say we want to build a car with a suspension with a certain amount of power. So we look at past failures. If we gather all the data and remove the data (?)

* Getting rid of missing/outlier data may cause serious errors, especially when the data is within a technical specification. If these outliers are valid performance expectations, we should include those in the final dataset
* **In terms of missing datavalues, if a column is bad, we can remove it, but it will remove that feature. You could add another column saying that this feature is missing**, and the algorithm can try to work with it
* You can use the other dataset to predict what the missing values would be. If you take advanced statistics they’ll tell you how to do this
* We specify the missing value as the target value and the machine will
* Amputation - replace the missing value with the mean median mode, some sort of aggregate value based on the column. THis may cause bias, but it will generally be mitigated
* Recap - delete row, delete feature, predict the missing value, fill in value with aggregate,
* We can also have clamps, if x is larger than A replace it with A, if smaller than B replace it with B…
* You can use standard deviation +- 3 standard deviations
* But what you do depends on the data
* Also you may rely on domain knowledge (some knowledge about where the data came from and where it will go)
* The one thing that is very important is when dealing with missing values is that you don’t what to fundamentally change the distribution
* If you draw a histogram and it has a modal bimodal distribution…
* If you modify it and the bimodal distribution goes away something went wrong

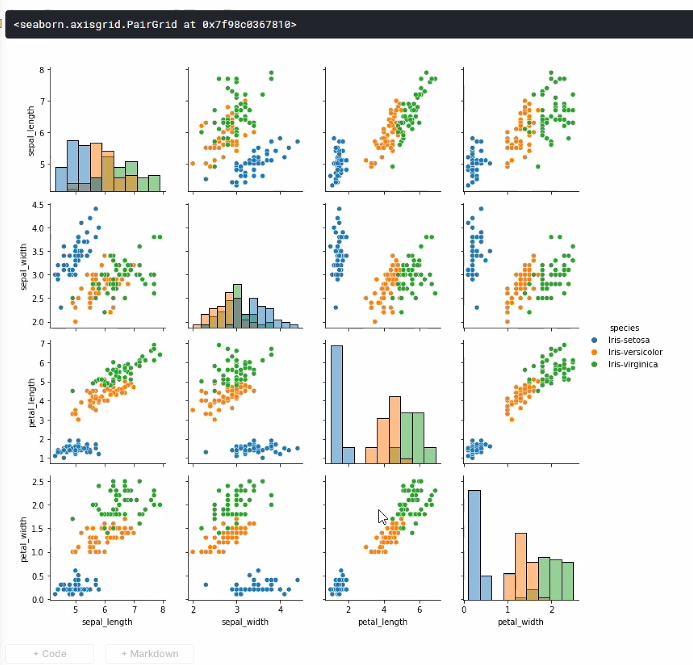
So really quick:

* Histograms are great, [camp?] charts are great
* Even more useful charts are scatter charts
* Multimodal distributions can indicate clusters, but you must look into the detail

Scatterplot:

* A scatterplot is just what you get when you plot one feature vs another feature
* Here we see that not only there are distinct vectors (?), but if we look at the species we can see a difference between green and red
* We can add in shape, size, and other features to create multidimensional data.
* Eventually this will become too hard to interpret

Scatter Matrix:

* Every feature plotted against every other feature
* This is used for figuring out are there relationships inside of the data
* Does one feature correlate with another feature?
* Using the seaborne library… we can do various things:
* 
* Looking at the data, notice some of the graphs can have a y=x line drawn on it to form the average
* The petal length and width seem to be correlated
* One of the reasons you want to do this is that if you can find a feature related to all the other features, you can remove it as a redundancy.
* We can see from the data that there is something we can “gather” from the data
* ScatterMatrices are very useful in finding correlations and making predictions
* If we look at uncorrelated data, it will look like noise (points everywhere, not making too much sense)

QUIZ PUSHED BACK TO THURSDAY, PROFESSOR WANTS TO COVER MORE STUFF ON TUESDAY.