Data Modeling

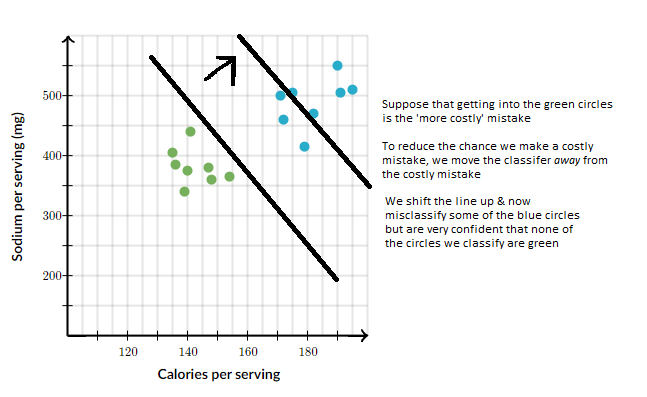
# Classifiers

## Definitions:

1. Hard Classifier: a classifier that is able to perfectly separate groups
2. Soft Classifier: a classifier that is unable to perfectly separate groups, but minimizes the number of ‘mistakes’ (misclassified) points
3. Structured Data - described/stored in numbers, can be quantitative or categorical
4. Unstructured Data – data is not easily described & stored (ex: text data)
5. Quantitative Data – numbers with meaning
6. Categorical Data – numbers without meaning (zip code, phone number), **non-numeric (hair color), binary (dummy, Y/N, On/Off, 1/0)**
   1. Binary data makes categorical data treated as quantitative
7. Unrelated Data – No relationship between consecutive data points (different customers or loan applicants)
8. Time Series Data – same data is recorded over time, usually recorded at equal intervals, relationship between data point to data point
9. **Data Point – all information about one observation NOT a single attribute of one observation**
10. SVM – Support Vector Machine (Model)
11. Scaling – put all data to some consistent interval, usually between 0 & 1
    1. Let
    2. **Must be in a bounded range (i.e. from 0:1)**
12. Standardization – scales each variable to a normal distribution, usually scale to standard normal (mean = 0 & )
    1. **Can be bounded or unbounded range**
13. KNN (K Nearest Neighbor) – assumes each point is ‘similar’ to other points near it, k selects the number of points ‘similar’

## Models

### Choosing a Classifier

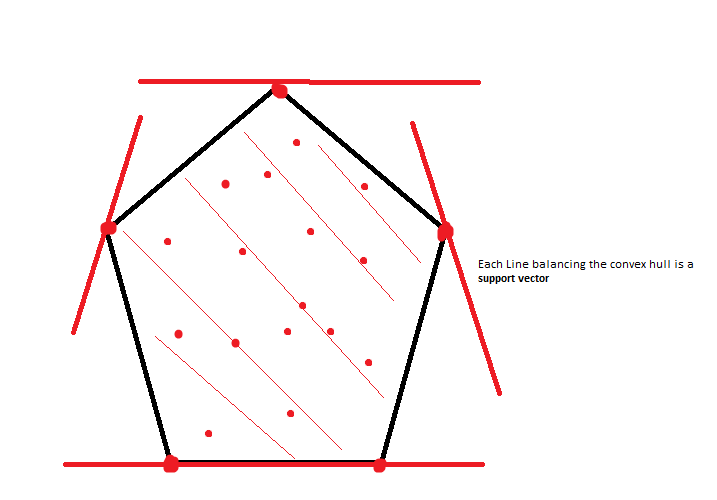
* A classifier can be ‘hard’ or ‘soft’ & separates groups of data points while minimizing the change of marginal error
* Technically, there can be infinite number of classifiers that separate groups (hard or soft), but we want a classifier that reduces the chance of error by increasing the distance the classifier is from each group
  + Minimize error by maximizing distance
* Some errors are more costly than others (cost of loan defaulting > cost of denying a good loan applicant)
  + When an error is more costly, we move the separator *away* from the costly errors
  + 
  + If separator is vertical or horizontal we only need 1 classifier
    - Suppose our classifier in the graph above was vertical, this would mean that no matter how much sodium an item has it will not change the amount of calories, therefore sodium does not impact the number of calories in food
    - If the classifier was horizontal, this would mean that no matter how many calories a food item has, it will not change the amount of sodium
    - **This means we can throw away one of the classifiers, in the first example we would throw away sodium (has no impact on calories)**

### Support Vector Machine Model (SVM)

* SVM Components:
  + N = number of data points (rows)
  + M = number of attributes (columns)
  + attribute (column) of the data point (row)
  + response for data point
    - Note, O is considered a good applicant for a loan, X is considered a bad applicant for a loan in this example
* We are creating a linear classification (does not have to be, but for example is linear)
  + Equation for a line is :
    - We can rewrite:
  + A *parallel* line is the same equation but with a different intercept ()
* For SVM we make a line that captures all of group 1 & a line that captures all of group 2
* This means that if our intercept = 1, any data point above this line will capture all of the “O”, if intercept = -1 then any data point below the line will capture all of the “X”
* If the line = 0, then the line is in the middle between each of the lines (1+(-1))/2 = 0
* We have Three Lines now (one that gets all the “O”, one that gets the “X”, and one exactly in the middle:
* Note, we want the middle between because this is the most distance between each line
* To get all of the O data points and all of the X data points, we can sum the area above & below the line
* Note, when this gets all of the O’s, when this gets all of the X’s
* Now we want the **distance** between the two parallel lines
  + Distance of parallel lines:
* We can drop the coefficient outside of the summation as it will not impact our model in the aggregate (we are making a line, so a scalar multiple of the line is actually the same line just stretched out)
* Remember, we want to **maximize** the margin by **minimizing** the distance between the parallel lines
* We call the following term the **margin denominator**
* To apply this to a soft classifier we will have to accept some level of error
  + ‘Correct side of Line’
  + ‘Wrong side of the line’ (i.e. measures the error)
  + We can summarize the error term as:
    - The error will either be 0 (we did it perfect) or the value will be a negative number (based on how is designed with an error
  + To **minimize** the error we **maximize** the following (remember error is negative so if we maximize the sum of negative numbers we get the smallest value of error)
* To Model the tradeoff between minimizing the error (reducing total mistakes) & maximizing the margin (being certain how next point is rightly classified
* **As increases we are accepting more error for higher levels of margin**

#### How SVM got its Name

* The previous example showed how to classify two groups with a linear line. Many times we want to classify multiple groups
* We can create a boundary around a group of points in a classification we would get the **convex hull**
* The **support vectors** are the lines that hold the shape up



* For the basic support vector machine model, we want 2 parallel lines that are as far apart as possible while still correctly classifying
  + **The parallel lines are the support vectors (in previous example**
* **Note, we want the line that maximizes the distance between the two lines**

#### Advanced SVM

* SVM finds a classifier that reduces error & enlarges margin between classifiers
* Hard separation can have multiple classifications, we may want a classifier that introduces error but ensures we are confident in our classification for future points
  + We model this by increasing the term
* In soft classification, we may add a penalty for misclassification
* We have added the term which is a penalty for **type** of error
  + If , margin gets bigger, use for more costly mistakes
  + If , margin gets smaller use for less costly mistakes
* Most times we need to scale or standardize data
  + When we scale data we place all attributes on the same interval (usually between 0 & 1)
    - Putting data as a percent is a way to scale data
    - Example of when to scale
      * Credit score is limited from 300 – 850 , income ranges from 0 – billions
      * A change in 1 point on a credit score could make a huge jump in income, we may not want this
  + Once data is scaled we can remove coefficients that aren’t needed (example, if a coefficient is near 0, it has almost no impact on the model & may not be necessary)
* SVM works similar in 2D to ND

### K-Nearest Neighbors Classification (KNN)

* KNN is able to classify multiple classes
* **Assumes each point is ‘similar’ to other points near it**
  + Example: income of $1,000/year & Credit Score of 500 is a ‘similar’ applicant to someone with income of $1,100 & Credit Score 550
* K = number of points ‘closest’ to
  + Can be any number, if K = 7, we look at 7 nearest points
  + **Note, we do not want to include the point itself in KNN**
* We find ‘closest’ using distance formula
  + Straight line distance:
  + Weighted straight line: :
* To find best number of K, will need to guess & check

# Data Validation, CV KNN, Kmeans

## Definitions

1. Real Patterns – real relationship between attribute & response
2. Random Patterns – looks like a real relationship but is not
3. Fitting Data – how the model optimizes based on the data points
4. Overfitting Data – model thinks random patterns are real
5. Training Data – develops model
6. Validating Data – Shows ‘best’ model
7. Test Data – Evaluates performance of model
8. Observed Performance – real quality + random effects in a model (usually too optimistic)
9. Splitting Data – usually split 70-90% of data to train/validate & remaining to test
10. Random Splitting – randomly choose observations for train, test, & validate
11. Rotation Splitting – take turns selecting points (commonly used when you have daily data where one day of the week is different from other days in the week)
12. Cross Validation – Split the data as usual, but keep iterating over the data so each data point is used in the train & validation portion
13. Clustering – grouping data points that are close to each other or similar
14. Euclidean Distance (straight line) –
15. Rectilinear Distance (ex city maps) –
16. Generalized (P-Norm) Distance –
17. Heuristic – not guaranteed best solution but usually gets very close to best solution & is done quickly
18. Outliers – datapoint that is very different from rest of datapoints, sometimes should be included (if a real datapoint) sometimes removed (if little chance of being reproduced or a data entry error)
19. Supervised Learning – when we know the response or outcome variable, uses **both** attributes & responses
20. Unsupervised Learning – when we do **not** know the response or outcome variable, uses only attributes

## Data Validation

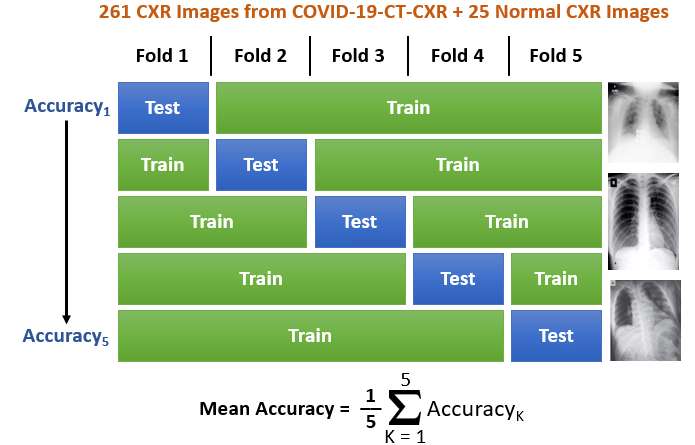
* Validation tells us how ‘good’ our model is
* When you fit a model to all of the data, this creates a ‘positive’ bias because you are fitting to both the real relationship between the data & the random patterns
* Each time you iterate through training & testing data you remove the random patterns in the data & get a more ‘true’ relationship between the datapoints
  + This is why the fit of training is greater than validate which is greater than test

## Choosing the Best Model

* We can train/test using different models & select the one that has the best performance
  + This is not a best practice method, the model that performed the ‘best’ may have done so by luck, to verify best model we introduce a third fold of validating the data
* Flow Chart
  + 1- Build Model(s) with **training data**
    - 1.A if choosing between multiple models Choose best model from **validation data**
  + 2- Estimate quality of model using **test data**

## Cross Validation

* Sometimes there are important data points in both the validation & test data, without using all of the data our model can’t account for these important points
  + This is overcome using *cross validation*
* Example



* We have 5 ‘folds’ use first 20% of data for test, 80% to train, Fold 2 moves to the next 20% & repeats, as we move through the 5 folds we increase the accuracy of the models. **We typically choose the model with the highest average**
* **Once we have chosen the best model we rerun on all of the training data**
* Summary: K-Fold Cross Validation is a better use of data, better estimates of model quality, & chooses model more effectively

## Clustering

* Clustering is group data points that are similar to each other
  + Commonly used in target marketing to segment similar customers with a specific message
  + Clustering helps you find groups you expect **and those you do not expect**

## K-Means

* Let = attribute of datapoint
* Let
* Let coordinate of of cluster center
* **We want to group data points that minimize the distance from the clusters center**
  + This way we feel confident that the groups of datapoints are ‘similar’ to each toher
* What this model is saying is we are **minimizing** the distance of the datapoint to the cluster center for all the datapoints & all datapoints can only be in 1 cluster
* This process takes a lot of computing time, so we use a kmeans heuristic that usually gets us close to the real answer very quickly

#### K-Means Algorithm

1. Pick k cluster centers within the range of the data
2. Assign each datapoint to nearest cluster
3. Not all the datapoints are in the center of the cluster, so we recalculate
   1. Repeat part 1 with new centers
   2. Repeat part 2 with new aligned data points
   3. Repeat a + b until no datapoints are moved

* K-means Algorithm is a form of Machine Learning (ML) & is a **heuristic**
* Kmeans assigns an outlier to whichever cluster is closest, **this does not mean that the outlier is similar to the group**
  + By removing the outlier, it prevents the cluster from being skewed towards the outlier
* Since K-Means is a heuristic, it is fast but not **always the best solution**, since it is fast, we can rerun several times testing different values of k
  + As k increases, the distance in each cluster gets smaller, but this does not mean it is better, theoretically you can make each datapoint in its own cluster, but is a bad model for predicting
* **Elbow Graph**
  + The elbow graph measures the total distance between clusters
  + You select where the kink (or elbow) in the graph is as the optimal number of clusters
  + The ‘kink’ is where the marginal benefit of adding 1 additional cluster decreases
* Predictive Clustering
  + After model & clusters are made, if a new datapoint tis entered there are two options:
    - 1- data point fits in existing cluster, datapoint goes into the cluster
    - 2- data point does not fit in existing cluster (now an outlier) & goes into closest cluster

#### Supervised vs. Unsupervised Learning

* Classification is used when we know the response (paid loan vs. did not pay loan) & is a form of **supervised learning**
* Clustering is when we do not know the right grouping (do not have a response variable) & must decide how to group & is a form of **unsupervised learning**
* Both clustering & classification are grouping methods

# Data Preparation, Outliers, & CUSUM

## Definitions

1. Outlier – datapoint that is very different from the rest of the data, can be a single point or multiple points
2. Outlier by Omission – when we expect an outlier at a certain interval but is missing. We know the range where the outlier should live but do not know the exact
3. Box & Whisker Plot – graphical method showing possible outliers
4. Logistic Regression – Model that can be used to estimate the probabilities of outliers occurring under different conditions
5. Change detection – detecting when a change has happened (usually in time series data)
6. Cumulative Sum (CUSUM) – answers ‘has the mean of the observed distribution gone beyond a critical level’, detects increases, decreases, or both

## Data Preparation

* Regression calls attributes predictors, classification calls attributes factors
* Important to scale or standardize data, also important to address outliers
* Highly correlated variables complicates our model & makes it harder to interpret results

## Outliers

* In time series, outliers look like ‘kinks’ in the graph
* Need to know the context of the outlier, sometimes the data point is in the range of the other datapoints but *when* it occurred makes it an outlier (a temperature of 60 degrees in summer ATL would be an outlier for *summer* but not an outlier for overall temps)
* Outlier by omission is when we are expecting an outlier to occur & it is missing
* Can find outliers using box & whisker plot
* We can find outliers seeing if there is a datapoint that has high level of error
  + (in example with the weather, the model would predict a high temperature for summer & have a lot of error for the 60 degree weather)
* Find the cause of the outlier
  + Is it bad data? Then we can throw it out
    - Can impute data for that specific point
  + If it is real data we need to decide if we want to include the outlier
* We expect outliers in large datasets
  + If normally distributed data, we will have 4% of the data outside of 2 standard deviations which are considered outliers
* Removing *real* outliers makes model too optimistic as the outliers will continue to occur
* Handling Outliers
  + Can construct a logistic regression model to estimate the probability of outliers occurring under different conditions
  + Build a 2nd model, 1st model has all the data, 2nd has the outliers removed, expect predictions somewhere in between model 1 & 2
  + **If real data has an unexpected outlier that are unlikely to be a repeatable occurrence it is a good idea to remove these datapoints**

## Change Detection

* Change detection catches when a change has happened in the data
  + Usually time series data (changes over time)
* Falls into 3 categories
  + 1- Whether action might be needed (global warming)
  + 2- Impact of past actions (black Friday sales YOY)
  + 3- Determine changes to help plan (Has voting patterns changed over time?)
* Change detection used to catch traffic buildups before they happen
* Railroad Axel heating up, if it gets too hot the train derails

## Cumulative Sum (CUSUM)

* CUSUM answers ‘has the mean of the observed distribution gone beyond a critical level
* Detects increases, decreases, or both
* or the average is the expected observation if there is no change

We want to know is

* Note,
  + is how far datapoint is above the average
  + is the value we got in the previous calculation
  + If we add it to the next calculation because we want to know if there is a positive change
  + If we make the value 0 because we don’t care if the value is lower than the previous
    - What this is saying that if the running total is below 0 (our average), then it will not help us detect an increase
* To make the model *less* sensitive we can add a value of C
* For our model to detect change now, must be **at least c higher to be included in detecting change**
  + There is tradeoff between levels of C & T of either being too conservative or too risky
* To find decrease flip equation:
* To find both, you do both increase & decrease at the same time

# Exponential Smoothing, ARIMA, GARCH, Moving Average

## Definitions

1. Trend – change in data over time not due to random fluctuation, can be increasing, decreasing, or cyclical
2. Randomness – changes in data that is not a trend and is randomly occurring
3. Exponential Smoothing – analyzes time series data using responses from many historical time periods, can be built to accommodate trends & seasonality (is a form of autoregression)
4. Stationary Data – data where the mean, variance & other values are constant over time
5. Differencing – method to find the differences between data points
6. Autoregression – predicting current value based on previous values, regression makes predictions based on Mom & Dad’s height, autoregression makes prediction based on childs height last year
7. ARIMA (Autoregressive Integrated Moving Average) – uses autoregression on differences
8. Moving Average – using the average value of certain time period to predict future values
9. GARCH ( Generalized Autoregressive Conditional Heterskedasticity) – estimate or forecast **variance**  of time series data

## Exponential Smoothing

#### Randomness

* Randomness is changes the data that is purely random & not from a trend
* Let = the expected baseline response at time period
* Let the observed response at time period
* We want to know:
  + 1- does (i.e. the observed response is the real indicator of the baseline)
  + 2 – does (i.e. today’s baseline is the same as yesterdays baseline)
* Exponential Smoothing Combines both 1 + 2
* If there is a lot of randomness, then we want so the baseline is equal to what it was yesterday
* If we believe that the observed value is the true value then we want

#### Trends

* In Exponential Smoothing, allows us to tradeoff how much we trust the newest piece of information
  + This **does not** account for trends or cycles
* To account for trends:
* Note
  + = times the difference in baselines
  + = times yesterday’s trend
  + Initial condition is that

#### Cyclical Trends

* Two methods to account for cyclical trends – additive or multiplicative
  + Additive works similar to regular trend
  + Multiplicative is typically used with seasonality’s

#### Seasonality

New baseline formula including trend & seasonality:

* Note
  + is the cyclic factor, by looking at the time period from L time periods ago we are comparing season to season (so 1st day of summer this year to 1st day of summer last year)
* Example: if c = 1.1 on Sunday, it means our sales were 10% higher purely because it was Sunday
  + If we sold 550 on Sunday, 500 would be from the baseline, 50 would be from c
* Sometimes called, single, double, triple exponential smoothing or HoltWinters

#### Exponential Smoothing Name

* Smoothing
  + Comes from the , terms
  + With we are ‘smoothing’ out the spikes in the data
    - If then we equally weight the new observation with our historical data
    - When the new observation is too low, the other half of the equation will pick up the new observation
    - When the new observation is too high , the other half of the equation pulls it down
* Exponential
  + Exponential smoothing incorporates all historical values into the most recent value, where new observations are weighted higher than historical
  + If you keep plugging in previous values of you get an exponential format
  + After some algebra:

#### Forecasting with Exponential Smoothing

* The further out you forecast with exponential smoothing the less accurate the forecast gets
* To forecast, we want to know
* We do not know , so our best guess is that
* Plugging in…
* Keep repeating….
* We can forecast to account for trend, our best guess for the next periods trend is
* Some idea for seasonality:
* We can use optimization to find the best values:
* Values of that minimize the squared error are the best values

## ARIMA Components

* We only explore at high level to see what ARIMA does, will not go into theory
* Three main components of ARIMA, differences, Autoregression, moving average

#### Differences

* Exponential smoothing works well if the data is stationary (mean, var are constant)
* Many times data is not stationary, but the *differences* in the data are stationary
  + Find the difference between observed value today – observed value yesterday
  + ….keep repeating
* Use **dth number of differences**

#### Autoregression

* Autoregression predicts current values based on previous time period’s values
  + In contrast, regression predicts new values based on other factors
* **Autoregression** **only works with time series data**
  + Exponential smoothing is an expanded form of autoregression (uses historicals to predict futures)
* ARIMA combines autoregression & differences
  + Uses the **p time periods** of previous observations to predict the dth order differences

#### Moving Average

* Assume that previous errors predicts future errors
* Order of moving average, goes back  **time periods**

## ARIMA Model

* Arima combines the P, D, & Q of the components:
* We choose the **dth order differences , pth order autoregression & quth order moving average**
* Exponential smoothing is the ARIMA(0,1,1)
  + Includes d & q, but no p\*\*\*\*\*\*
* Applications of ARIMA
  + ARIMA(0,0,0) – models ‘white noise’ , no pattern
  + ARIMA(0,1,0) – models ‘random walk’
  + ARIMA(p,0,0) – models only autoregressive
  + ARIMA(0,0,q) – models only moving average……
* ARIMA is powerful because it generalizes many simpler models
  + Tends to work better than exponential smoothing for stable data
  + Rule of thumb is you need at least 40 data points

## GARCH Model

* Generalized Autoregressive Conditional Heteroskedasticity
* Estimates or forecasts variance of time series data
  + Variance lets us estimate the level of error
  + Used in investments, balances expected return with amount of volatility in data
* GARCH & ARIMA are very similar
* How GARCH is different from ARIMA
  + 1) GARCH uses variance & squared error
    - ARIMA uses observations & linear errors
  + 2) GARCH uses raw variance
    - ARIMA uses the difference of variance
* Since GARCH doesn’t use differences, it only needs p & q

# Linear Regression, AIC, BIC, & Box-Cox

## Definitions

1. Simple Linear Regression – linear regression with 1 predictive variable
2. Multivariable Linear Regression – linear regression with more than 1 predictive variable, answers descriptive & predictive questions
3. Likelihood – measure probability density for any parameter set
4. Maximum Likelihood – parameter that gives the highest probability
5. Akaike Information Criterion (AIC) – method that assess if model would improve by removing predictive variables, the smaller the AIC the more likely less variables will be better
6. Bayesian Information Criterion (BIC) – similar method to AIC, but is more conservative than AIC (favors even less predictive variables), can only be used of number of datapoints> number of predictors
7. Descriptive Component – part of regression that shows how each predictive variable impacts the response variable
8. Predictive Component – part of regression that allows you to predict values based on given inputs
9. Causation – 1 thing causes another thing to happen, **cause comes before effect, makes intuitive sense, & no outside factors are causing the effect**
10. Correlation – 2 things happen to move together in some pattern, **does not imply causation**
11. Box-Cox – method that transforms the predictive variables, the response variable, or both to better fit the data, use QQ plot to see if box-cox is needed (if errors are normally distributed then it is NOT needed)
12. Error – difference between observed & predicted value
13. Sum of Squared Errors – is the distance between the predicted value & the observed value, is a measure of how good the model is
14. Confidence Interval – where the probability that the coefficient value would occur if you redid the regression with a new dataset
15. T-Statistic – the coefficient value divided by the standard errors, is related to P-Value
16. P-value – H0 says that the coefficient is = 0, H1 says coefficient is =/ 0, the smaller the p-value the higher likelihood that the coefficient is not equal to 0
17. R-Square – estimates the variability of the model, if R-Square = 60%, the model explains 60% of the relationship between the response & predictors & 40% of the relationship is unexplained by the model
18. Adjusted R-Square – same as R-Square but adjusts for the number of predictors in the equation (R-Square has a bias to increase as predictors increase, this corrects for that bias)

## Simple & Multi Linear Regression

* Simple Linear Regression, is regression with 1 predictor variable
  + Predictors = attributes, outcome = response
* SLR looks for a *linear* relationship between predictors & outcome
* Basic SLR =
* Multivariable Regression has additional predictor variables:
* Measure how good the model is using the SSE
* Error =
* Line of best fit, minimizes the sum of squared error

## Maximum Likelihood

* We measure the model’s quality of fit using maximum likelihood
* Likelihood assumes that the data we have is correct, calculates the variance, then calculates the probability density that the model will calculate the value that are produced
  + Likelihood measures probability density for all predictors in the model
  + **Maximum Likelihood (ML) is a parameter that gives the highest probability**
* We can use likelihood to compare different models
  + Use each models likelihood ration then perform a hypothesis test
* ML Fitting
  + Simple regression assumes each data pointis independent (iid) & the errors are normally distributed

#### Akaike Information Criterion (AIC)

* AIC is a form of Maximum Likelihood
* Where,
* is the ‘penalty term’ which balances likelihood with simplicity & helps prevent overfitting
* Models with smaller AIC are better because this means *less variables increase likelihood*
* AIC assumes there are infinite number of data points (which we never have in data) so there is a correction term to account for this
* Note
  + is the correction term
* We can compare the output of AIC & , if both values are similar we can proceed using the simple AIC without the correction term
  + This is because similar values mean they have similar likelihood of occurring, so the correction does not benefit

#### Bayesian Information Criterion (BIC)

* BIC is similar to AIC but encourages even less parameters than AIC
* You can **only use BIC if n>k**

* Notice, as n approaches the value of k, the formula fails, which is why n must be greater than k to work
* The closer n is to k BIC will not work as well
* Rule of thumb for interpreting BIC
  + If , smaller model is **very likely** to be better
  + If , smaller model is **likely** to be better
  + If , smaller model is **somewhat likely** to be better
  + If , smaller model is **slightly likely** to be better
* Best practice is to use AIC, BIC & ML together and determine which is the best metric for your problem

## Using Regression

* Regression answers **descriptive & predictive questions**
* A coefficient in a regression is interpreted as: **on average, predictor variable x1 increases (decreases) the outcome variable by (value of a1) units**
* To use regression predictively you build your model with the data, then use the calculated model to predict using the new datapoint
* **Regression is not prescriptive**
  + Prescriptive is random walks, trees + branching, & transformation

## Causation vs. Correlation

* Causation is when one thing causes another
  + Cause comes before effect
  + Idea of causation makes sense
  + No outside factors causing effect
  + Typically we can suggest causation but this is difficult to prove
* Correlation is when 2 things happen to move together in some pattern (**does not imply causation**)
* Very difficult to prove causation

## Transformations

* Linear Regression is good if there is a linear relationship between the predictors & the outcome variable
* If the data is not linear, there may be another type of relationship that would be better (quadradic, exponential, log, etc.)
* We can transform the predictors, the response, or transform both
* Box-Cox transformation is automated to do this in statistical software
* There can also be interactions between predictor variables
  + Example: to estimate the childs height, might use mom’s height, dad’s height, & both mom & dad’s height together (multiple x1 & x2)
  + For this example the multiplied variable says, as both parents height increases (decreases) it impacts the predicted childs height by (coefficient value)

## Regression Output

* Regression uses hypothesis testing on each predictor variable:
* As the p-value decreases, the higher likelihood that we **reject the null hypothesis**, so we have higher beliefs that the coefficient value of the predictor variable is not equal to 0
* Standard threshold is if the p-value is > .05 this is a good variable, if it is less then it is a bad variable
* When you have big data, p-values become very small even when the values aren’t related to the response
* P-values are probabilities even when they are meaningful (very small)
  + Example, if I have 100x’s & each p-value is .02, each predictor has a 2% chance of being insignificant
    - This means that by probabilities, 2 of my variables are irrelevant to the model
* Confidence Interval is where the coefficient probability lies
* T statistic is the coefficient divided by the standard error & is related to the p-value
* Sometimes a coefficient value will be very small even if it is significant (meaning it is likely this is the true value of the coefficient but has almost no impact on the predictor variable.)
* R Square estimates the variability in the model, adjusted R-square corrects for using additional predictor variables

# PCA, Box Cox, & Detrending data

## Definitions

1. Box-Cox – transforms response variable to remove heteroskedasticity
2. Logarithmic Transformation – transforms nonnormal distributions to a more normal distribution
3. QQ Plot – plot that tests if the error in the model is normally distributed, works by transforming the response variable to remove heteroskedasticity
4. Detrending Data – used typically for time series analysis, can detrend response variable, predictors, or factor-based models
5. Principal Component Analysis – method that removes correlation between predictor variables, also ranks coordinates by importance, helps reduce the effect of randomness & higher ranked PC are likely to have higher signal:noise ratio
6. Eigen Vector - is a value that transforms a square matrix to be orthogonal
7. Eigen Value – the scalar value that makes the eigenvector

## Box Cox Transformation

* Some models have assumptions that data is normally distributed, if this assumption is wrong, then the results are biased
* Heteroskedasticity – when the variance is not constant (normally distributed variance is close to constant)
  + When variance is heteroskedastic in regression, the higher variance creates higher weights on the far right side which can make the coefficient estimate larger. This increases the error because the model is trying to fit to these higher values
* Logarithmic transformation stretches out smaller range to enlarge variability & shringks larger range to reduce variability
* Box cox works by finding the best value to get a response vector to a normal distribution
* We can use a QQ plot to see if a box cox is needed

## Detrending Data

* Trend in time series data can mess up factor based analysis (like regression)
  + Prices are going up each year, but part of that is inflation so you need to take inflation out to see the impact of ‘real’ prices
* You can detrend:
  + Response variables
  + Predictors
  + Factor-based models (SVM, regression, etc.)

## Principal Component Analysis

* Use PCA when you want to know which factors are most important to include
* You need lots of data & some predictors may be highly correlated with each other
* PCA
  + Removes correlations within data &
  + Ranks coordinates by importance
    - Focusing on the highest ranked PC this:
      * Reduces effect of randomness
      * Higher ranked PC likely to have higher signal-to-noise ratio
  + PCA reduces number of factors & removes correlation between factors
  + **This is not used as variable selection because each new factor is a linear combination of the original factors**
* In order to interpret coefficients in regression after PCA, you must transform the variables back

## Eigenvalues & Eignevectors

* You use Eigenvectors to transform a square matrix into a matrix that is orthogonal to the original matrix
  + When two matrices are orthogonal to each other they have 0 correlations to each other (size of determinate is 0)

# Trees, Random Forests, & Logit Models

## Definitions

1. CART – classification of models called Classification And Regression Trees
2. Decision Trees – type of tree that says, if X happens I choose A, if Y happens I choose B
3. Root Node – the starting place of a Tree
4. Branch – each split from a node
5. Leaf – when a branch has ended
6. Pruning – after tree model has been built, cutting off least important branch & rerunning model
7. Random Forest – algorithm randomly makes hundreds of decision trees & averages outcomes, reduces randomness in comparison to a single tree
8. Logistic Regression – response variable is 0 or 1 & the coefficients estimate probability of 0 or 1 occurring
9. Sensitivity – True Positive / (True Positive + False Negative), Percentage of correctly classified positives, is the ‘True Positive Rate’
10. Specificity – True Negative/(True Negative + False Positive), Percentage of correctly classified negatives is the ‘True Negative Rate’
11. Receiver Operating Characteristic (ROC) Curve – plots specificity versus sensitivity (True positives vs. true negatives), smaller values on the xaxis indicator lower false positives & higher true negatives, higher values on the yaxis indicate higher true positives & lower false negatives
12. Area Under Curve (AUC) – probability that model estimates a random ‘yest’ point higher than a random ‘no’ point
    1. If AUC is .5, then there is a 50/50 chance of either of outcome
13. Confusion Matrix
    1. Matrix that captures True Positive, False Positive, False Negative, & True Negative
14. True Positive – the model correctly identifies a positive case
15. True Negative – the model correctly identifies a negative case
16. False Positive – the model incorrectly identifies a positive case (i.e. the case is really negative)
17. False Negative – the model incorrectly identifies a negative case (i.e. the case is really positive)
18. Regression Spline – spline is a function of polynomials that connects to each other (there are kinks or ‘knots’ between polynomial to polynomial)
19. Bayes Regression – regression that starts with data & an estimate of how the regression, coefficient, & random error is distributed (estimate given from expert opinion, good when you have little data)
20. KNN Regression – does not estimate predictors, you plot all the data, and the predict response is the average response of K closest data points

## Trees in Regression

* Trees can be used in either classification or regression
* In a single regression modeling you may have a 0/1 outcome our a continuous response you are trying to predict
* Decision Tree is a tree that determines a yes/no outcome

#### Trees with binary Response Variables

* Suppose you are trying to predict who should get a marketing email
* Linear regression would use all of your data points
* You may believe that the distributions of the response variables are different from certain age categories (younger behaves differently from older)
* You branch between young & old
  + Run a regression on each leaf, then find coefficients for that model
  + If you have 10 leaves, you will have 10 regression models that only incorporate specific slices of the data

## Branching

* Two questions when branching
  + Which factors to branch?
  + How to split data
* Best practice is to branch 1 factor at a time
* Each time you split you run a regression then try split again, if variance decreased split again, keep repeating until decrease in variance is less than the threshold
  + The more splits in a tree higher likelihood of overfitting
* If a branch makes low improvement in model, then prune

## Random Forest

* Random forest introduces randomness into the model
* Creates many trees & averages the outcome, the average of many trees may be more accurate than a single tree
* **If using regression then you use the average of predicted responses**
* **If using classification use the mode of predicted response**
* Random Forest provides better estimates by reducing overfitting
* Random Forest are harder to explain output/interpret, doesn’t explain how variance intereacts between variables
* Gives an aggregate measure than a specific
* Works as a black box predictor but not good for detailed insight

## Logistic Regression

* Similar to linear regression, but outcome variable is a 0/1 & the coefficients are percentage increase/decrease of likelihood of outcome being 1
* We can perform logit trees but they take a lot of processing time
* Can you ROC to plot specificity & sensitivity
* ROC gives a quick & dirty estimate of the models quality

## Confusion Matrix

* Plots True & False Positive/Negative

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model Prediction | | |
| True Classification |  | Yes | No |
| Yes | Correct, True Positive | Wrong, False Negative |
| No | Wrong, False Positve | Correct, True Negative |

* Can use a confusion matrix to evaluate the quality of model
* Can estimate total cost of wrong/correct tradeoffs

# Variable Selection Methods

## General

* There are two reasons to limit number of factors in model:
  + 1- Overfitting (happens as m -> n or m >n)
    - Too many factors in model may fit to random effects & make bad estimates
  + Simple is preferred to complex
    - Less data is required, less chance to incorporate irrelevant data, easier to interpret

## Forward, Backwards, & Stepwise Selection

Note: ‘Good Enough’ can be based on pvalue, rsquare, AIC, BIC, or other fitting metrics

**Greedy algorithm -** algorithm where each step in process adds/removes one things & evaluates

### Forward Selection

* Start with 0 factors, add with best new factor, test if factor improves model (can be p value or r square value), add another factor...etc.
* Fit model with current set of factors, remove high pvalues, fit model with final set of factors

### Backwards Elimination

* Start with all the factors, find worst factor, was factor ‘bad’ enough, remove...etc.
* Remove any with final factors with high p value, fit final model

### Stepwise Regression

* Combo of forward & backward elimination
* Start with no factors, find best new, if factor is ‘good enough’, add another factor, fit model with factors, remove high p value factors, ‘do we have enough factors’, find best new....
* Once final best new factor is added, remove high p value variables, fit model with final

## Lasso, Elastic Net, & Ridge

### General

All three methods add a constraint to the standard linear regression that creates a ‘budget’ for the total size of all coefficients. This method limits the number of variables to the the ‘budget’ tau while also minimizing the SSE from normal linear regression

You must scale all of these methods, otherwise large values will have very large coefficients & eat up the whole budget

Different levels of Tau gives trade offs on: # of variables, quality of model, & best tradeoff\*

Lasso uses sum of absolute value, Ridge uses sum of squared values & Elastic Net combines both of these using a lambda variable that weights either the lasso or ridge constraint

### LASSO

* Same as Linear Regression with a constraint limiting the total size of all coefficient values
* **Effective of selecting Variables**
* **If variables are highly correlated, may select the ‘wrong’ one (based on intuition)**
* **Before using LASSO need to check correlations of variables, may preemptively remove highly correlated variables to the preferred on**

### Ridge

* Constraint uses sum of squared coefficients
* Ridge tends to have coefficients shrink toward 0 & reduces variance in estimate, this is because the quadratic constraint
  + Shrinking coefficients increases bias but lowers variance
  + Prediction error is a function of bias & variance, if you reduce variance by enough (even though increase bias) you can improve the model
* **Effective in improving prediction**
* **May underestimate coefficients of very predictive variables**

### Elastic Net

* Adds another constraint combination of absolute value of coefficients + their squares
* **Receives both benefits of LASSO & ridge**
* **Receives both negatives**

# Design of Experiments (DOE)

## Definitions

1. Compare - look at same dimension for like items (price on red car vs. price on blue car)
2. Control - make sure dimensions are similar to each other (make/model/etc.), need to have all data on dimensions for controls
3. Blocking - way of encompassing many factors in 1 dimension (sports cars are probably similar & family cars are similar, then compare colors of each type)
4. A/B Testing - way to choose between 2 different alternatives
5. Full Factorial Design - Tests every combination of A/B Testing
6. Fractional Factorial Design - Tests a subset of all A/B Combinations
7. Independent Factors - factors that are all independent of each other
8. Exploration - more information
9. Exploitation - immediate value
10. Bernoulli Distribution - 1/0 outcome
11. Geometric Distribution - probability of having x Bernoulli Failures until 1st success
12. Poisson Distribution - distribution over intervals of time
13. Weibul Distribution - Time between failures
14. QQ Plot - Can show distribution of data
15. Queing - model oComparen how to optimize a queue or line
16. Memoryless property - historical values do not matter
17. Simulation - build model to watch behavior
18. Deterministic Simulation - No randomness (same inputs -> same outputs)
19. Stochastic Simulation - system has Randomness
20. Continuous Time - changes happen continuously
21. Discrete Event Simulation - changes happen at discrete times
22. Discrete Event Stochastic Simulation - use when system have high variance (randomness) & the average is not good enough (must be precise)

## Compare , Control, Blocking

|  |  |  |
| --- | --- | --- |
| Type | Definition | Example |
| Compare | Look at **same** dimension for like items | Price on red car versus blue car |
| Control | Make sure same dimensions are **similar** | Red & blue car are same make/model/same safety features/etc |
| Blocking | Grouping many factors into 1 dimension | Compare sports cars to sports cars because they have |

## A/B Testing

* Used when comparing two groups (is X better than Y? Is Average of XX different from Average of YY?)
* Example: suppose two adds show 2000 times & we track which is clicked:

|  |  |  |  |
| --- | --- | --- | --- |
| AD | clicks | uses | % |
| A | 46 | 1003 | 5% |
| B | 97 | 997 | 10% |

* Use hypothesis testing to see if A is better than B
  + Can do hypothesis testing each time an ad is clicked & once one is highly significant better than another, stop testing
* **AB Testing Criteria**
  + **1- collect data quickly**
  + **2- data must be representative**
  + **3- data collected must be small compared to total population**
    - **Otherwise there is little benefit to hypothesis testing as we have the ‘truth’**

## Factorial Design

* Full - Use when you have multiple variations of hypothesis testing you can test **every** combination
* Partial - When too many combinations focus on a subset, using a ‘balanced’ design is each choice & each pair of choices appear the same number of times
* **Use anova test to determine importance of each combination**

## Independent Factors

* If we believe the controls are independent then we can use regression, depending on factors can include interaction terms (ex: font + color are not independent)
* Helpful when doing before modeling or collecting data (so you don’t have to do experiments twice)

## Exploration, Exploitation, & Multi-Armed Bandit

* Slot machine is called a 1 armed bandit (because the expected utility of the player is negative)
* Many slot machines is called multi armed bandit
* Test with K alternatives, equal probability of selecting each alternative, repeat
  + Explore - choose alternative test on probability of each being the best
  + Exploit - after test, update probability of each being the best

## Probability Distributions

* A probability distribution gives us a guide for certain types of data
* It is good when we only have the response variable or collecting data is hard

### Bernoulli Distribution

* Typical example is flipping a coin, a y/n outcome
* Most useful when putting multiple Bernoulli Distributions together
* **Answers the question: probability of getting x successes out of n independent identically distributed Bernoulli Trials**
* **As n -> infinity, Bernoulli converges to normal distribution (normal distribution is useful for modelling errors)**

### Geometric Distribution

* **Answers the question: having x Bernoulli Failures until 1st success**
  + How many rejects till 1 accept
* Can use Geometric Distribution to test if IID is true (if response variable fits geometric then IID & you don’t need to collect any other data)
  + Ex: airport screening, are screeners more likely to pull someone aside if there hasn’t been a search in a while?
    - If geometric/IID then screeners treat people independently
    - If not geometric then they do not treat people independently

### Poisson Distribution

* Answers the question: How many of XX do we expect in this interval of time
* If Poisson, the arrivals are IID (one arrival doesn’t depend on another)
* If arrivals are Poisson, then time between successive arrivals is exponential distribution

### Weibul Distribution

* Weibul Distribution is similar to Poisson, gives the **time between failures**
  + Answers the question **How long until XX fails**
* When K < 1 then weibul good at modeling situations where the failure rate decreases over time (mechanical parts, if defective will fail immediately)
* When K > 1 then weibul good at modeling sitatutions where failure rate increases over time (car tires)
* When K = 1, weibul good for constant failure rate, which reduces down to Poisson Distribution

### Queuing

* Answers a question on how many employees we should have based on call volume
* Is considered memoryless, meaning whatever happened in the past with calls does not impact future calls
* Queuing Model parameters
  + Arrival Distribution, Service Distribution, Number of servers, size of queue, population size, & queue distribution

### Simulation

Is used to model/watch behavior, you can add various factors like randomness to see how good model is

Prescriptive simulation is good for ‘what if’ questions

Simulation can be overused & overtrusted (many times missing info or have bad info)

Telemarketer example, each employee has large variation that is hard to model

# Missing Data

## General

Missing data can happen from mechanical failure, forms incorrectly filled out, unknown, or purposely not given

Data issues come in as either missing or wrong data, usually there are patterns as to why it is missing or wrong

## Patterns in missing data

There may be a bias in the missing data, for example in liver transplants looking at how long lived after heart transplant will consider all living patients as missing

## Dealing with Missing Data

* We can throw away data
* Indicate value is missing
  + Can use binary, or interaction term
    - Interaction have both Missing & # of kids for missing gender
    - Using binary effectively makes 2 models in 1 (like a tree) compares all the values were there is no missing value & then compares all values were there are missing values
  + Look at the type of variables that are missing, there may be a bias involved
* Or estimate missing value (imputation)
  + Approach 1 - Midrange Value
    - Mean/Median/Mode (use Mean/Median for numeric & Mode for categorical)
    - This approach is easy to implement & hedges against being ‘too’ wrong (put int some central tendency value)
    - This can create a bias in imputation (suppose all high values omit & keep putting in average value of lower values, will bias towards lower values)
  + Approach 2 - Regression
    - Attempts to reduce/eliminate bias, creates a regression to predict the missing values & then use those values to run next model
    - More accurate than midrange
    - Is complex, is a lot of effort, & does not capture all variability
      * Can add perturbation to add variability
  + Approach 3 - Regression + Perturbation
    - **Do not impute for more than 5% per factor**
    - **Imputation means you use the data to fit twice, so if you impute too much can lead to overfitting**
    - New Model Error = Imputation Error + Perturbation Error + Model Error
    - Hard to account or quantify all the error, can use test/train data

# Optimization

## General

* Most models we have looked at so far involve some form of optimization (anything with a min/max is a form of optimization)
* Both descriptive analytics & predictive analytics feed into optimization (prescriptive analytics)
* Three components of optimization:
  + Variables (optimization software will pick)
  + Constraints + Hidden Constraints (we must pick & can be very difficult)
  + Objective function
* A solution in optimization are values/coefficients for each variable
* A Feasible solution is one that satisfies all constraints, an optimal solution is the **best solution**
* Binary variables allow for more complex models, but increases the amount of time it takes to optimize
* If objective function is concave (max) or convex (min) & constraint x is a convex set, then it is easy to solve, but may have a long solution time **(called a convex optimization problem**)
* Linear Program limited to Integer Values very difficult to solve (called **integer problem**)
* General Non-Convex Problem is the most difficult to solve
* Most optimization problems have 2 steps:
  + 1- create a 1st solution
    - Can be bad/simple/infeasible
  + 2- find an improving direction t, use step theta to move along t
    - New solution = old solution + theta
  + 3- Repeat until solution doesn’t change much or time runs out
  + **Note, if this is a convex optimization problem we are guaranteed to find the optimal solution, if it is nonconvex we may find a local max but not global max**

## Types of Optimization Problems:

1. Regression (minimize SSE)
2. Lasso (Regression + Constraint)
3. Ridge (Regression + Constraint)
4. Elastic Net (Regression + Constraint)
5. Logistic Regression (objective function is to maximize likelihood of parameters & minimize predicted error)
6. SVM Hard Classifier - Constraints is each point is correctly classified, objective function is to maximize the margin
7. SVM Soft Classifier - no constraints, objective function is to minimize number of misclassified points while maximizing margin
8. Exponential Smoothing - Alpha, Beta, Gama are constraints (all between 0 & 1) and objective function is to minimize error
9. ARIMA - no constraints, objective function is to minimize error
10. Clustering - Variables are centers for cluster centers, constraints is only 1 cluster per datapoint, objective function is to minimize the distance between cluster centers & datapoints

#### Easiest to Hardest Optimization Problems to Solve:

1. Linear
2. Convex Quadratic
3. Convex
4. Integer
5. General non Convex

## Network & Neural Network Problems & Deep Learning

* Network Model is a type of Linear program
* Objective function is a linear function of variables, constraint flow into one node must equal flow out of node, flow on arcs between nodes can have a min/max set
* Examples of Network problems
  + Shortest path possible (google maps)
  + Assignment Model (which worker to which job maxes efficiency)
  + Maximum Flow (how much oil can come from pipe)

#### Neural Network

* Neural Networks & Deep Learning is a way to react to patterns we don’t understand
* Neural Networks trains system to patterns we can’t write a math equation for
* 3 levels in Network
  + Input Layer
  + Hidden Layer
  + Output
  + Each Neuron gets inputs from previous layer, calculates function of weight inputs, gives its output to next layer
* With enough data the weights settle on a value & has high predictive power (in practice requires a lot of data & don’t give good results

#### Deep Learning

* Similar to neural networks, but many layers (‘deep’)
* In practice seems to work better than neural networks, used for natural language processing, speech recognition & image recognition

# Nonparametric Methods

* **Use Nonparametric Method when: 1) don’t know distribution, 2) distribution doesn’t fit a nice form, 3) want to know about median, 4)don’t have much data**

## McNemar’s Test

* Compares results on pairs of responses
  + Example two competing treatments for virus
  + Q: is Treatment A > Treatment B?
  + Is a **binomial distribution can use this to calculate p values**
* Scenario 1: 32 neither worked, 7 only B worked, 61 both worked
  + P = .02, B>A
* Scenario 2: 12 neither worked, 27 B only, 20 A only, 41 both
  + P = .38, Treatment A is not necessarily better than treatment B

## Wilcox Signed Rank Test for Medians

* Assumptions: 1) Distribution is Continuous, 2) Distribution is symmetric
* Question: is the Median of Distribution different from M?
* Method: given responses
  + 1- Rank from small to large
  + 2- (sum of all ranks where
  + 3- p-value test for w, if p is small than median is likely different from m
* Method 2: Given Pairs of responses
  + Use z instead of m
  + Repeat method 1, if p-value is small then two sets of data are likely different

## Mann-Whitney Test

* Answers Question: Are two datasets different?
* Uses two datasets, but **not paired samples (assume all obs are independent)**
* Given independent observations
  + Rank all observations together
  + U = smaller of two adjusted rank sums:
  + Find significance of u

## Bayesian Model

* Uses conditional probability called Bayes’ Rule
* Example: True Positives = 98%, False Positives = 8%, 1% of population has disease, 8.9% of everyone tested tests positive
  + P(B|A) = .98
  + P(A) = .01
  + P(B) = .089
* If someone tests positive what is the probability they really have the disease? 11%, that is because the number of false positives outweigh the number of true positives

## Empirical Bayesian Model

* Use when the overall distribution is known (expert opinion) but only little data is known

## Interconnections in Populations

* Focuses on automating ways of finding highly interconnected sub populations
* Answers questions like how quickly will others get the flu given someone is positive, how quickly social media can disseminate information
* **Communities** is a set of circles that is highly connected with its, a **Clique** is a set of nodes that all have edges together

### Louvain Algorithm

* Decomposes graph into clique, then finds the max modularity of the graph
* Modularity measures how ell graph is separated into communities of modules
* Very similar to clustering method, is a heuristic so it gives a solution very quickly

# Game Theory

* Game theory is cooperative decision Making
* Timing can be simultaneous or sequential
* Strategy is best response to a problem, a counter strategy is best strategy to next move, etc.
* Sequential Game can have pure or mixed strategy
* Perfect info assumes we know everything about everyone’s information
* Imperfect info says some have more info than others (not symmetric)
* Zero-sum game- whatever 1 person gets the other person loses
* Non Zero Sum Game - Total Benefit can increase for all parties (but does not mean increases equally)