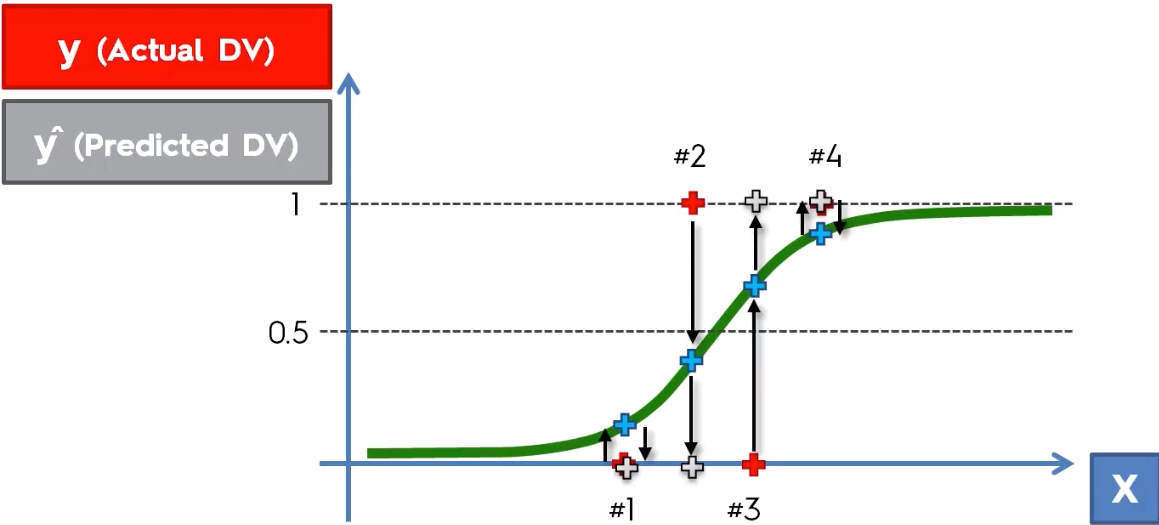
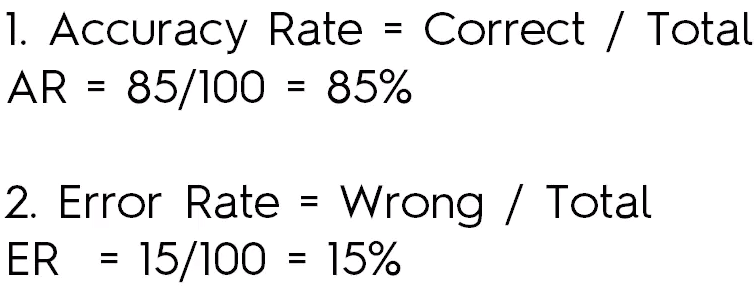
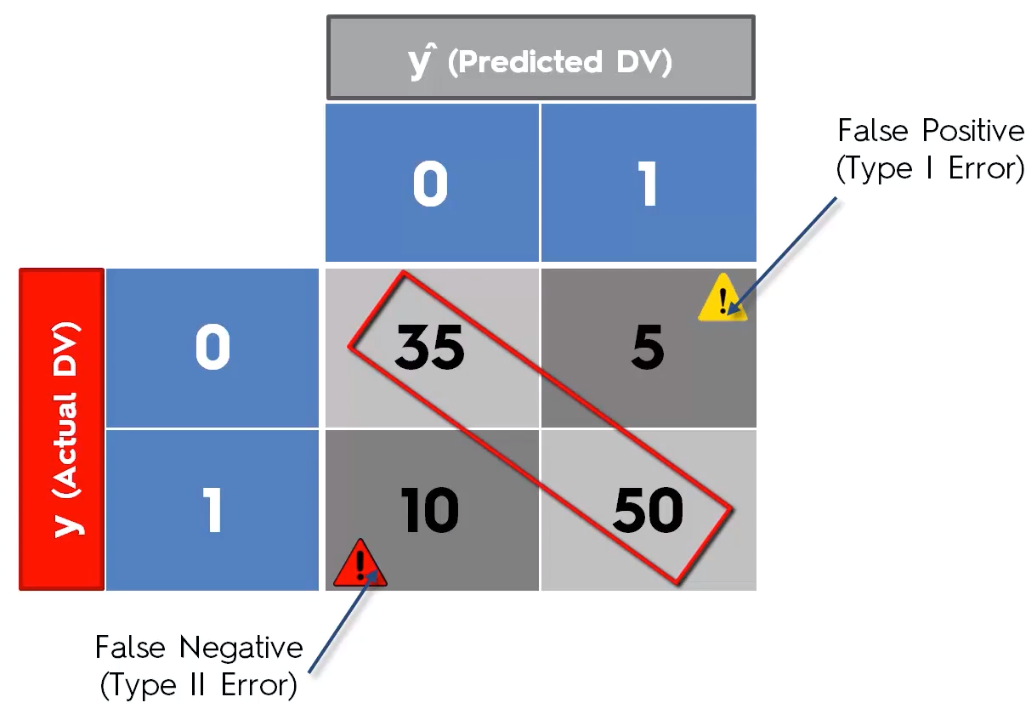
**Evaluating Classification Models**

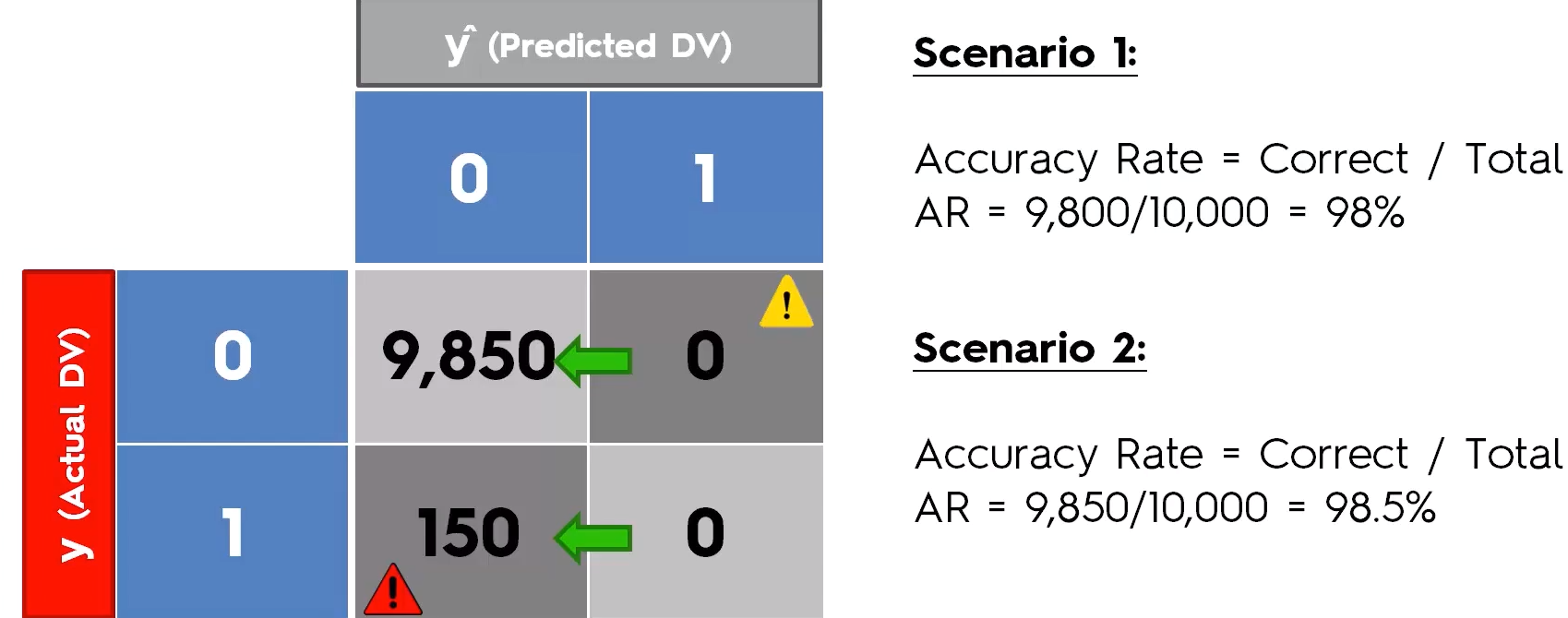
False Positives + False Negatives

* 
* Observations 1 + 4 were classified correctly and observations 2 + 3 were classified incorrectly
* 2 = FN (Type II error = predicted (-) but actually (+))
* 3 = FP (Type I error = predicted (+) but actually (-))
* Some think of Type I as “less dangerous”

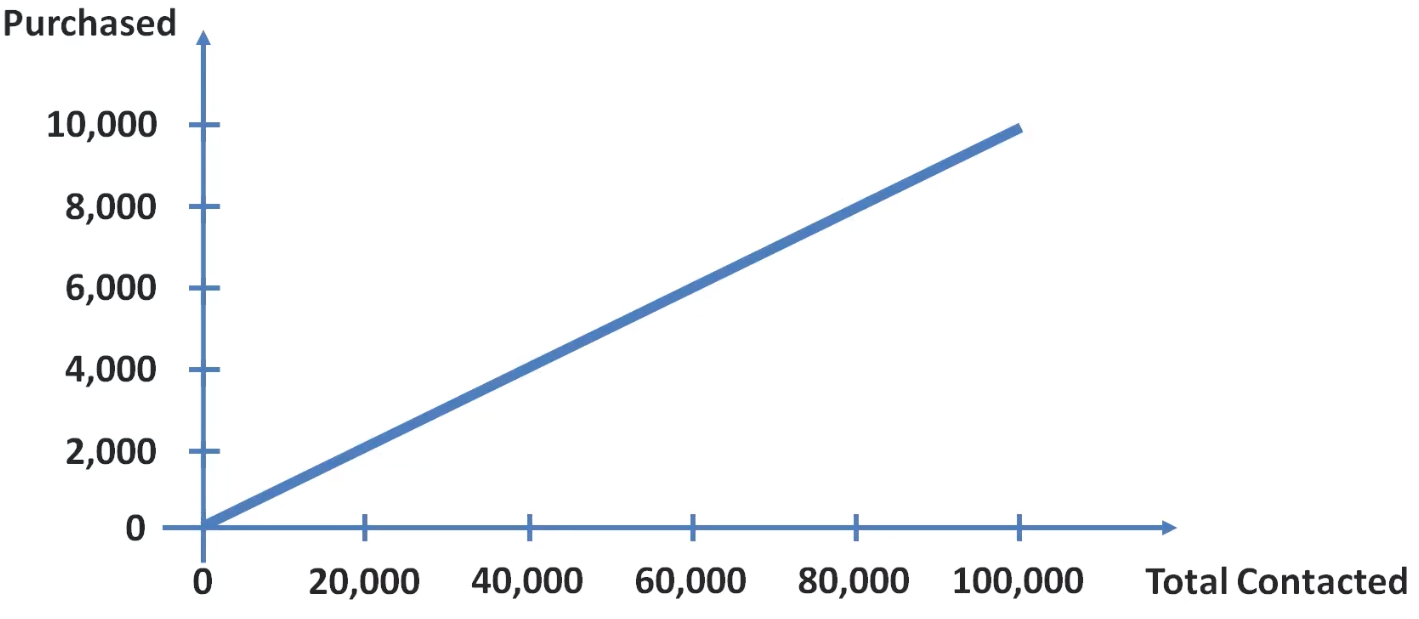
Confusion Matrix

* 

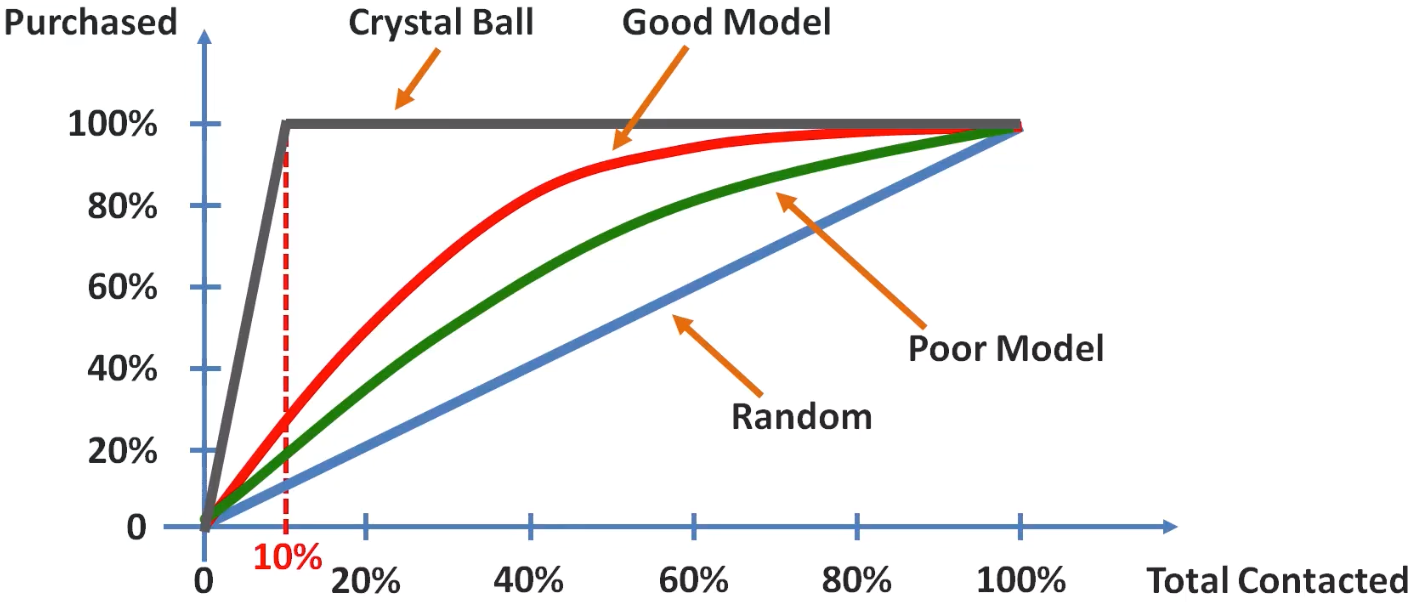
Accuracy Paradox

* Don’t base judgment of model “goodness” just on accuracy rate, as some things can go wrongs.
* Ex: 98% accuracy w/ a model, but then decide to just predict “0” for everything
* ,
* Accuracy actually *went up,* even though we have no actually logic in our classification

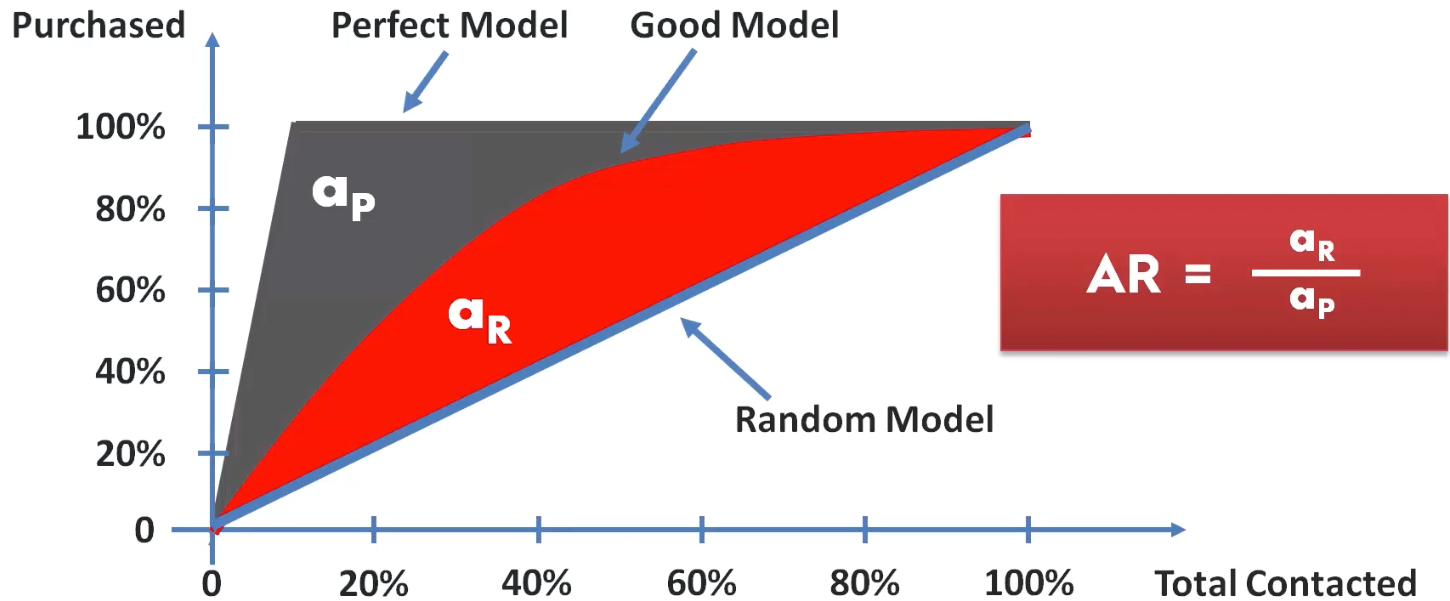
Cumulative Accuracy Profile (CAP) Curve



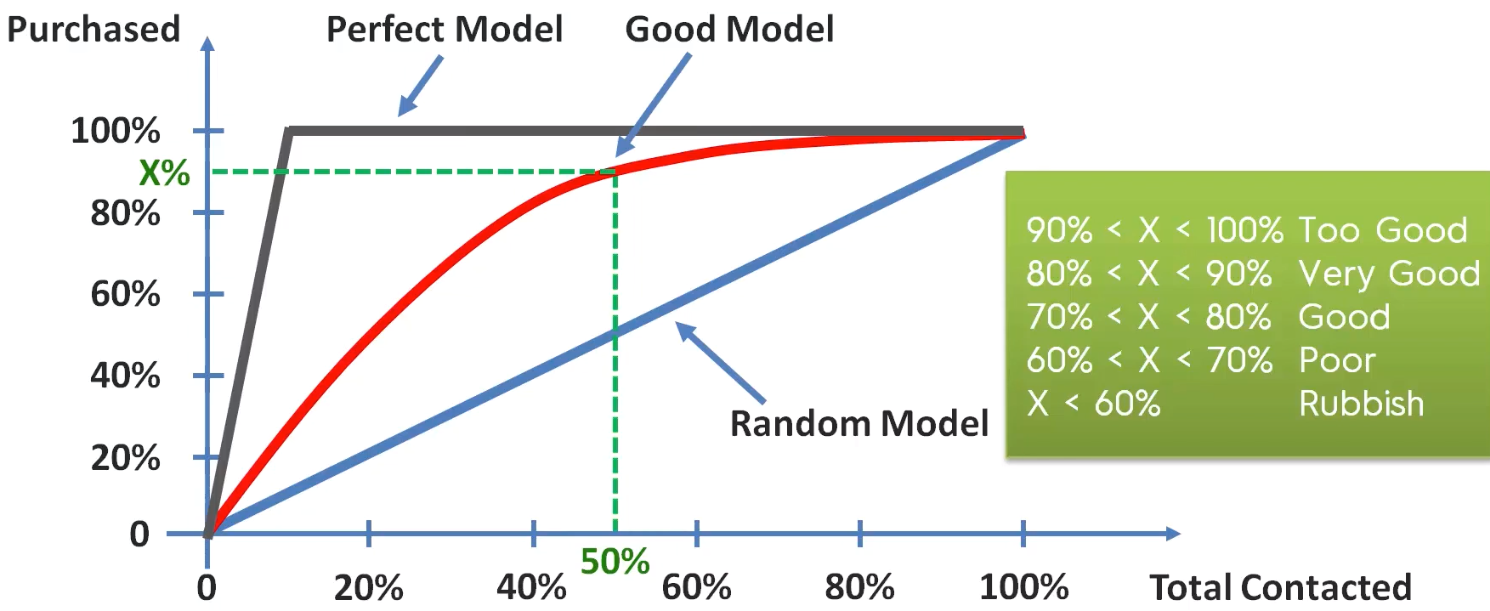
* Can we get more customers to respond to ads we send out? Can we target customers more appropriately to get a better response rate?
* Instead of sending letters to random customer samples (like above), we pick + choose customers to send to via a customer segmentation model



* As model “gets better”, area between curves increases + we can compare different models (red vs. green vs. blue) vs. the ideal line (where we know only a max of 10% of customers we contact will respond, so best case = contact 10% of customers and all of them respond)
* To quantify the accuracy ratio w/ these curves, we find the area under the ideal curve and the area under our “good model” and divide the good model area by the ideal model area



* Obviously values closer to 1 are better
* A second approach is to asses this visually, since calculating this without software is difficult.
* We look at the 50% line on the x-axis and look where it crosses the model and the ideal curve



* If a problem is linear, go for **Logistic Regression** or **SVM**.
* If a problem is non-linear, go for **K-NN, Naive Bayes, Decision Tree** or **Random Forest**.
* From a business point of view, you’d rather use:
* Logistic Regression or Naive Bayes when you want to rank predictions by probability.
* Ex: rank customers from highest probability they buy a certain product to lowest probability.
* Eventually allows you to target marketing campaigns (logistic regression if problem is linear + Naive Bayes if non-linear\_
* SVM when you want to predict to which segment customers belong to.
* Segments can be any kind
* Decision Tree when you want to have clear interpretation of your model results,
* Random Forest when just looking for high performance w/ less need for interpretation.
* Improve models w/ **Parameter Tuning** ofparameters that’re learnt (ex: coefficients in Linear Regression)
* **Hyperparameters** = parameters that are NOT learnt + are fixed values inside the equations of the models (ex: regularization parameter lambda or the penalty parameter )
* Finding optimal values for these = exactly what Parameter Tuning is about.