Machine Learning Applications and Methods

Numeric Prediction
Generalized Linear Models
Regression Trees

Different Approaches => Different Models

- **What to Optimize**
  - Minimize Prediction Error
  - Minimize Classification Errors
  - Maximize Probabilities
- **How to Find Parameters**
  - Search space of solutions
  - Constraints and Assumptions
- **What kinds of functions to use**
  - E.g. Linear vs NonLinear
  - E.g. divide input into pieces

Varieties of Regression

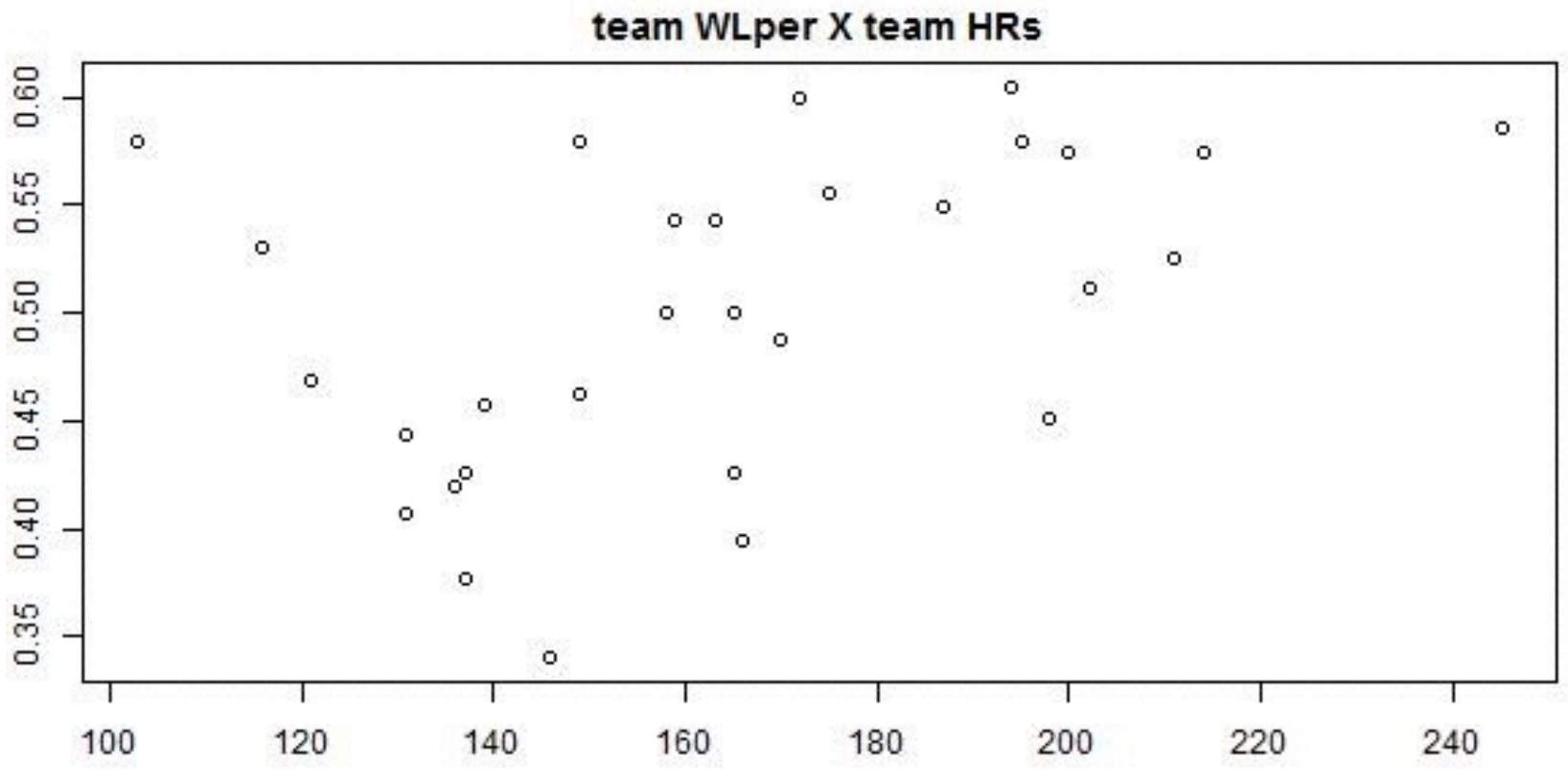
- **Linear Model:** $Y = X * B$

  where Y=outcomes , X=data matrix

  - Solve for B directly by algebraic manipulation
  - Solve for B directly by setting derivatives to 0
  - Solve for B iteratively by derivatives and small changes (that decrease error)
  - Solve for B but reweight by variance in X
  - Solve for B but constrain size

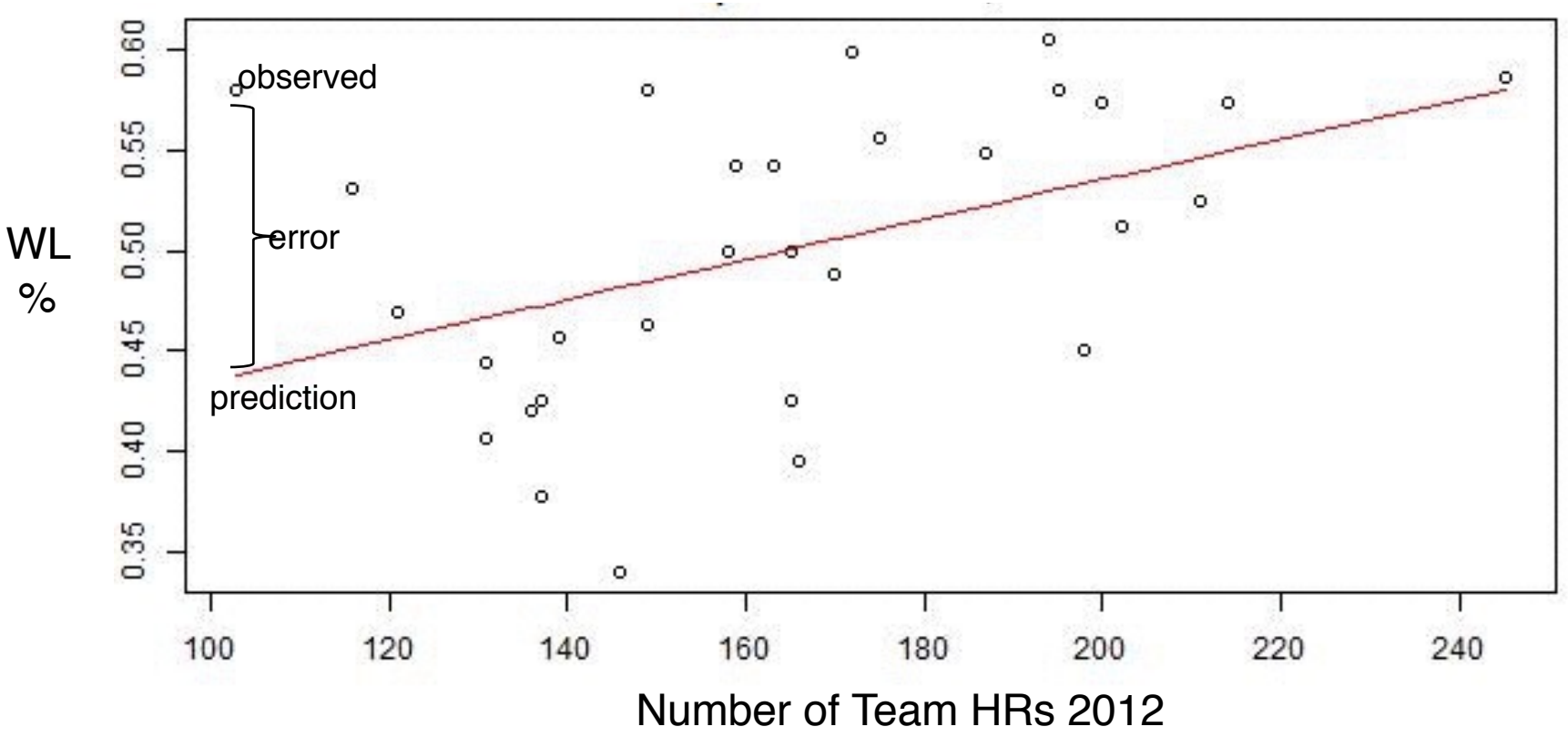
  …

Data Example: Home Runs and W-L



Linear Regression Model

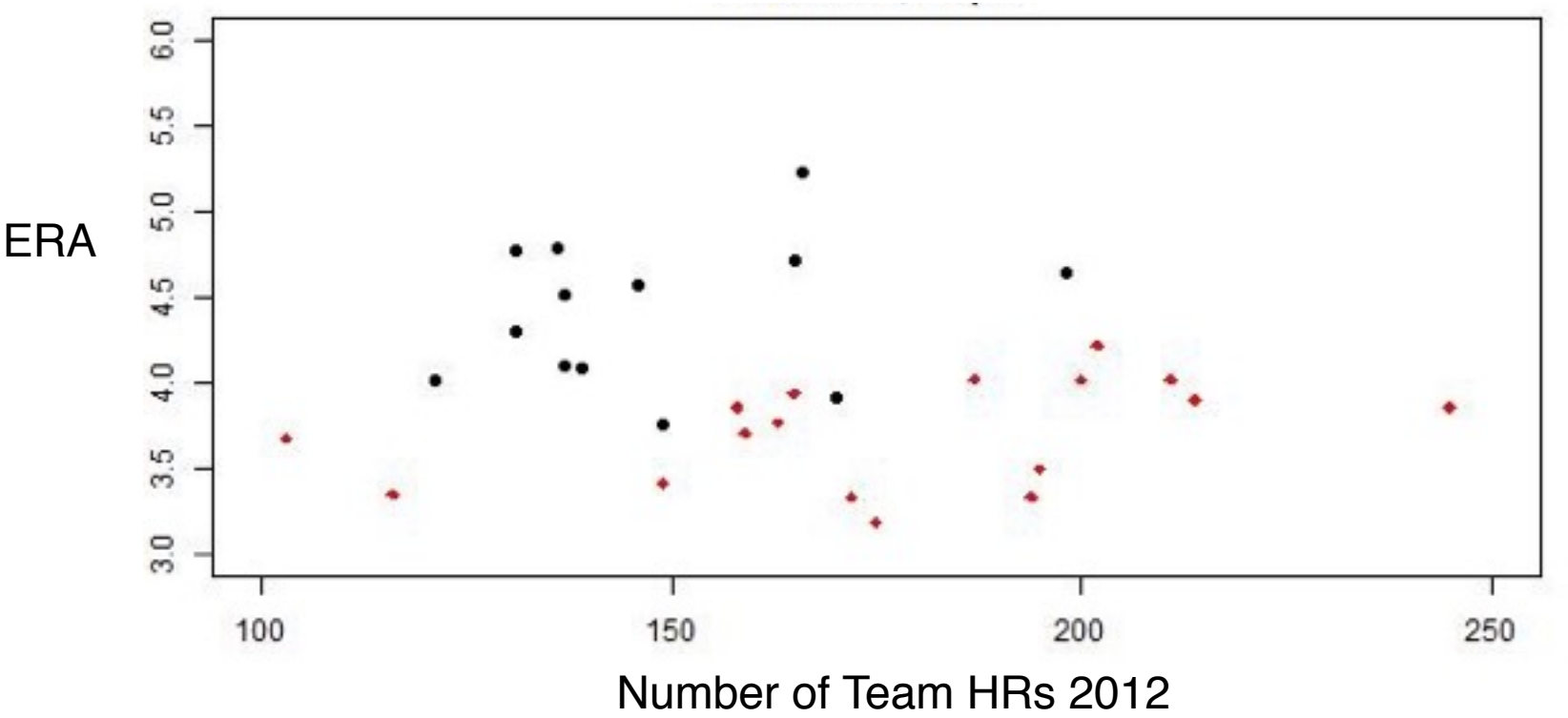
Q: What is the relationship between HRs and Winning %?



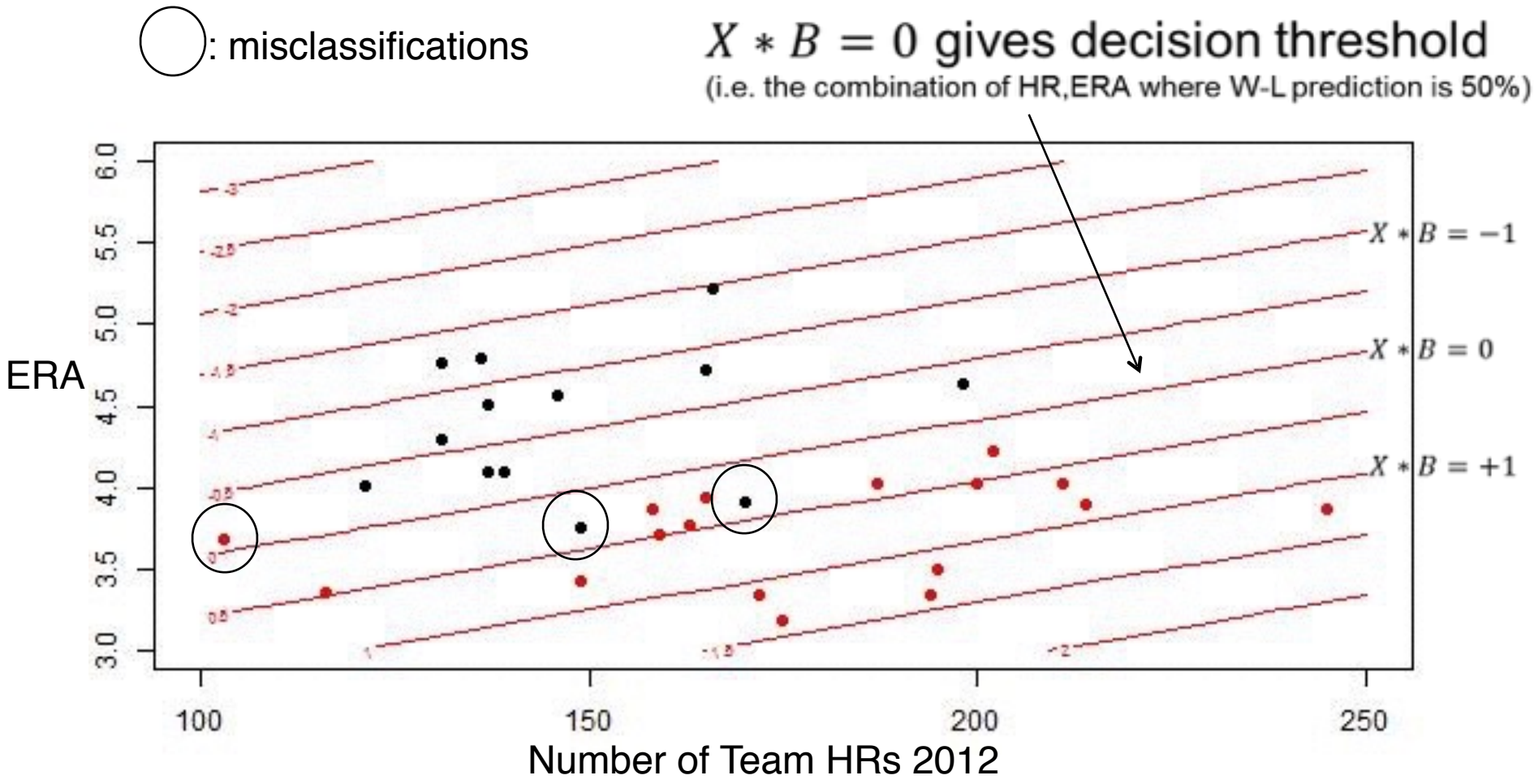
A Linear Model for Classification

- **target is 1 (WL%>=.5) and -1 (WL%<.5)**

Q: Can you classify winning records based on HRs and ERA?

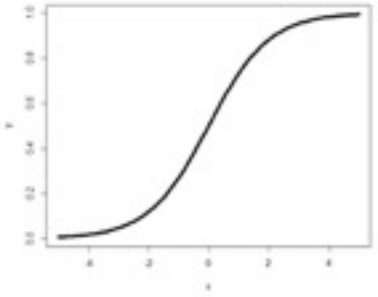


Linear Model for Classification (cont')



Linear to Logistic Regression

- **Squash X*B to 0,1 range using Logistic Function:**



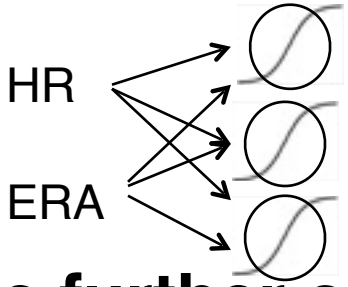
- **Directly Model P(YIX)**

  e.g. Probability(Y=Winning given HR,ERA values)

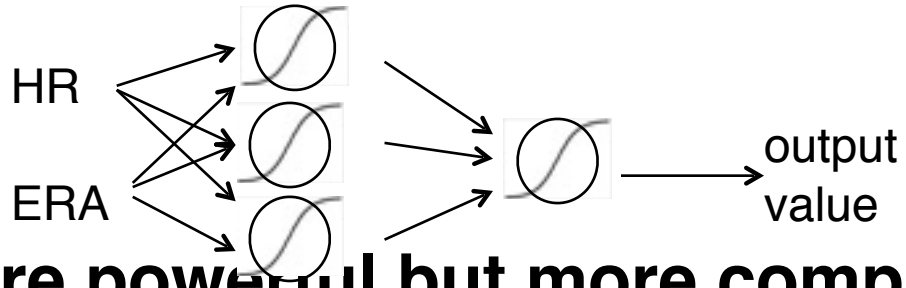
- **Solve with maximization methods**

Logistic Regression to Neural Networks

- **Use several squash functions (hidden layer)**



- **Take further combinations (output layer)**

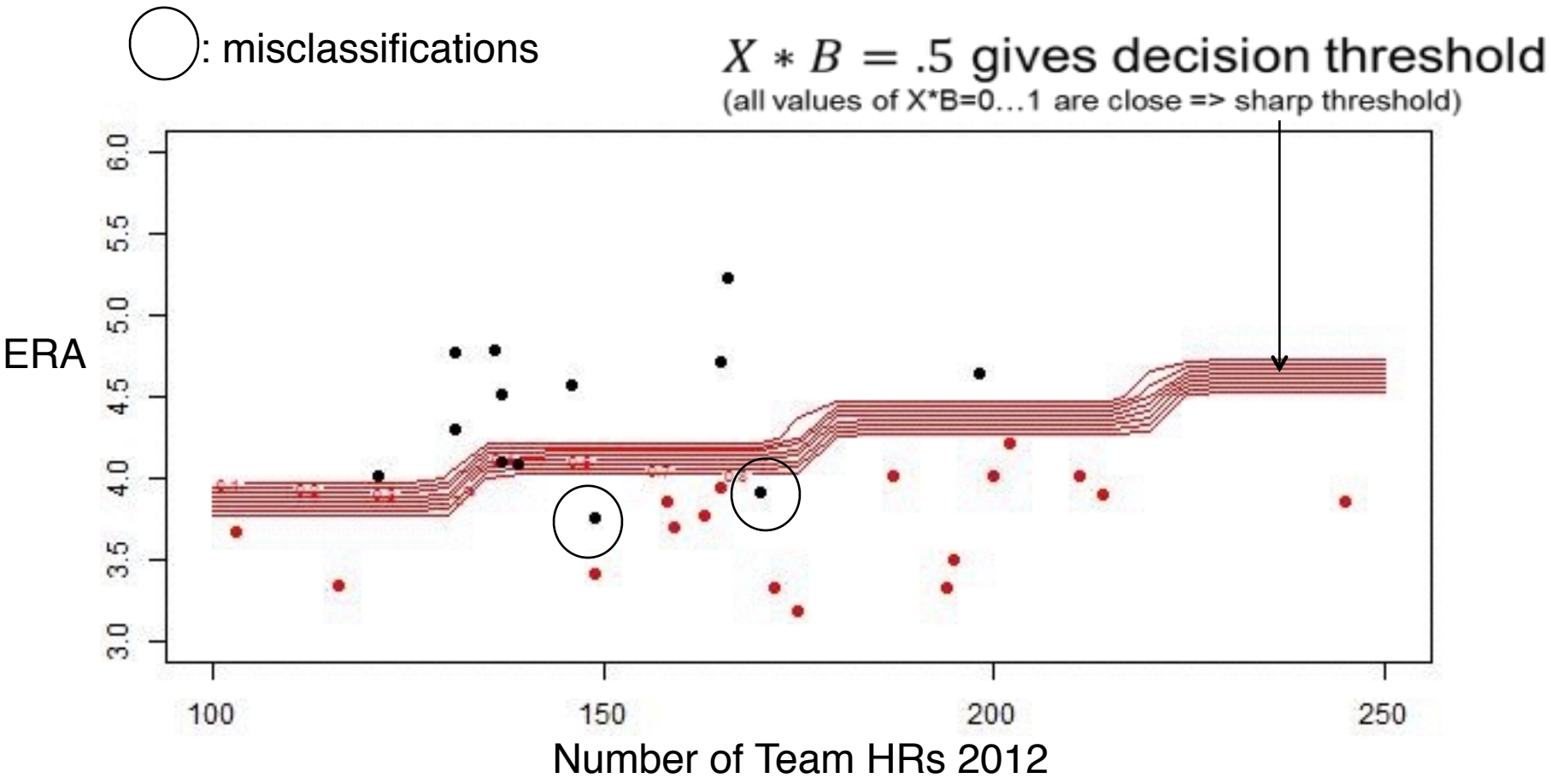


- **More powerful but more complex**

  many parameters, many options, needs more training

Neural Network classification

(R nnet() with 8 hidden units, 100 training iterations)

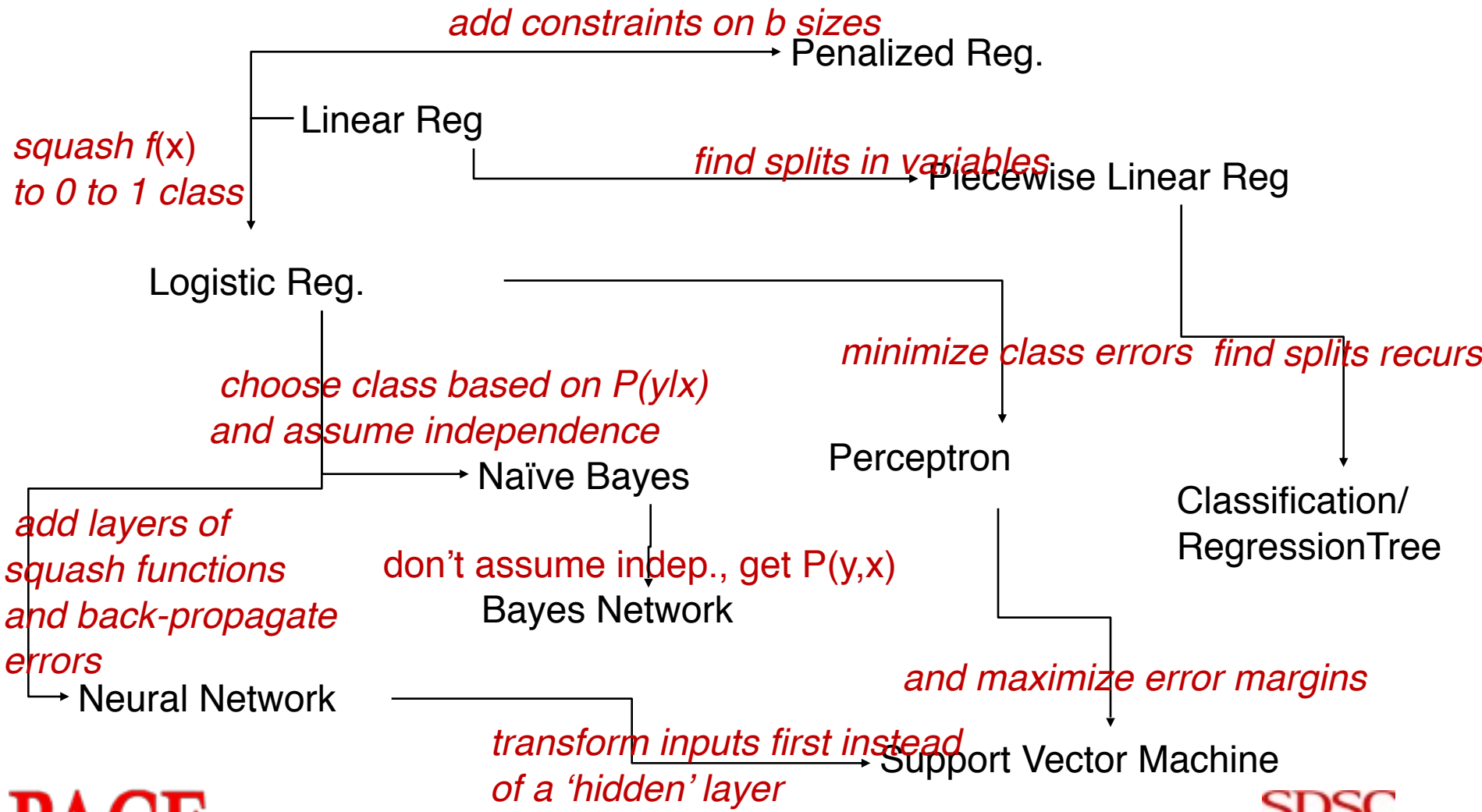


Other NonLinear Options:

- **Polynomial and multiplicative variable interaction**
  - add new variables: HR*HR, HR/ERA, etc..

- **Divide and conquer:**
  - Condition model on cut points (“knots”)
    - e.g. HR<140, HR>140
  - Splines or Trees (smooth or not smooth predictions around cuts)

Model Space Map
(one view on some parts)



The Big Picture of Models

	Function	Target	Error	Parameter Estimation
Linear Reg.	*linear*	*numeric*	*squared residual*	*Solve for least sq.*
Logistic Reg.	*nonlinear*	*prob. of class*	*misclass.*	*max likelihood*
Neural net.	*nonlinear*	*Numeric or Class*	*squared resid. X-entropy*	*Gradient descent*
Trees (classification and regression)	*piecewise*	*Numeric or Class*	*squared resid. Or misclass.*	*Greedy search and prune*
Support vector	*linear or nonlinear*	*Class*	*margin of misclass*	*Constrained optimization*
PCA,PLS	*Dim. reduction*	*Data reproduce*	*squared resid.*	*solve*
kNearest Nbr	*Local means*	*Numeric or class*	*squared resid. or misclass*	*Xvalidation on k*
…				

The Big Picture of Models

	Function	Target	Error	Parameter Estimation
Splines	*Polynomial, interactions*	*numeric*	*squared error*	*Solve w/ contraints*
Bayesian	*Use Prob. Dens.Fnctns*	*prob. of data and parameters*	*(un)expected values*	*Expectation max., Monte Carlo Markov Chain*
Ridge Regression	*linear*	*numeric*	*squared resid.*	*Solve w/penalty*
Matrix Factor	*linear*	*Data reproduction*	*squared resid. w/ penalty*	*iterate*
Lasso	*linear*	*numeric*	*absolute resid*	*Iterate in steps*
Perceptron	*linear*	*Class separation*	*Min classification*	*iterate*
Linear Discriminant	*linear*	*Class separation*	*Min variances*	*Matrix solution*
…				

Other Model Areas

- **Time Series**
  - i.e. linear regression with time steps
- **Network Analysis/Graphs**
  - i.e. model links and properties over whole graph
- **Many techniques combine**
  - e.g. Bayesian SVM, Bayesian network
  - e.g. Elastic Net (Lin. Reg. w/abs. & squared error)
- **Nonlinear Kernel transformations**
  - Kernel PCA, Kernal PLS, Kernel Ridge regression
- **Penalty can be added**
  - e.g. minimum norm in neural nets

Different Considerations => Different Model Choices

- **Data and Problem Issues to Consider:**
  - Dimension reduction to Factors?
  - Sparsity (in data or response)?
  - Multiresponse (prediction)/Multiclass (classification)?
  - Number of Observations vs. Variables?
  - Interpretability vs Model Complexity?
  - Bias vs Variance trade-offs (good vs stable estimates)
  - Scale

App Recommendations

- **Usually no absolute choice and no silver bullets**
  (otherwise we wouldn't be here)
- **Start with simple methods**
- **Consider trade off as you go more complex**
- **Find similar application examples**
  (what works in this domain)
- **Find paradigmatic examples for models**
  (what works for this model)
- **Goals and Expectations!**

Numeric Predictions

Numeric Predictions

- **Numeric prediction is interpreted as prediction of a continuous class**
  - Like classification learning but with numeric “class”
- **Counterparts exist for all schemes**
  - Decision trees, rule learners, SVMs, etc.
- **Almost all classification schemes can be applied to regression problems using discretization**
  - Discretize the class into intervals
  - Predict weighted average of interval midpoints
  - Weight according to class probabilities
- **Learning is supervised**
  - Scheme is being provided with target value

Numeric Prediction

- **Example: modified version of weather data**

Outlook	Temperature	Humidity	Windy	Play (time
Sunny	*85*	*85*	*False*	*5*
Sunny	*80*	*90*	*True*	*0*
.	*.*	*.*	*.*	*.*

CPU Performance Data

	Cycle Time	Main Memory (Kb)		Cache	Channels		Perform
	MYCT	*MMIN*	*MMAX*	*CACH*	*CHMIN*	*CHMAZ*	*PRP*
1	*125*	*256*	*6000*	*256*	*16*	*128*	*198*
2	*29*	*8000*	*32000*	*32*	*8*	*32*	*269*
.							
208	*480*	*512*	*8000*	*32*	*0*	*0*	*67*
209	*480*	*1000*	*4000*	*0*	*0*	*0*	*45*

Classifiers for Numeric Prediction

- **Schemes for numeric prediction include: linear regression, model tree generators, locally weighted regression, instance-based learners, decision tables, multi-layer perceptron**

- **Classifiers available in WEKA for Numeric prediction**
  - LinearRegression: (functions) linear regression
    - The simplest is linear regression
  - m5.M5Prime: model trees
    - M5Prime is a rational reconstruction of Quinlan's M5 model tree inducer
  - IBk: k-nearest neighbor learner
  - LWR: Locally Weighted Regression
    - LWR is an implementation of a more sophisticated learning scheme for numeric prediction, using locally weighted regression
  - RegressionByDiscretization: uses categorical classifiers

Numeric Prediction in Python

- **Available methods in**
  - Statsmodels
  - Scikit-learn
  - NumPy

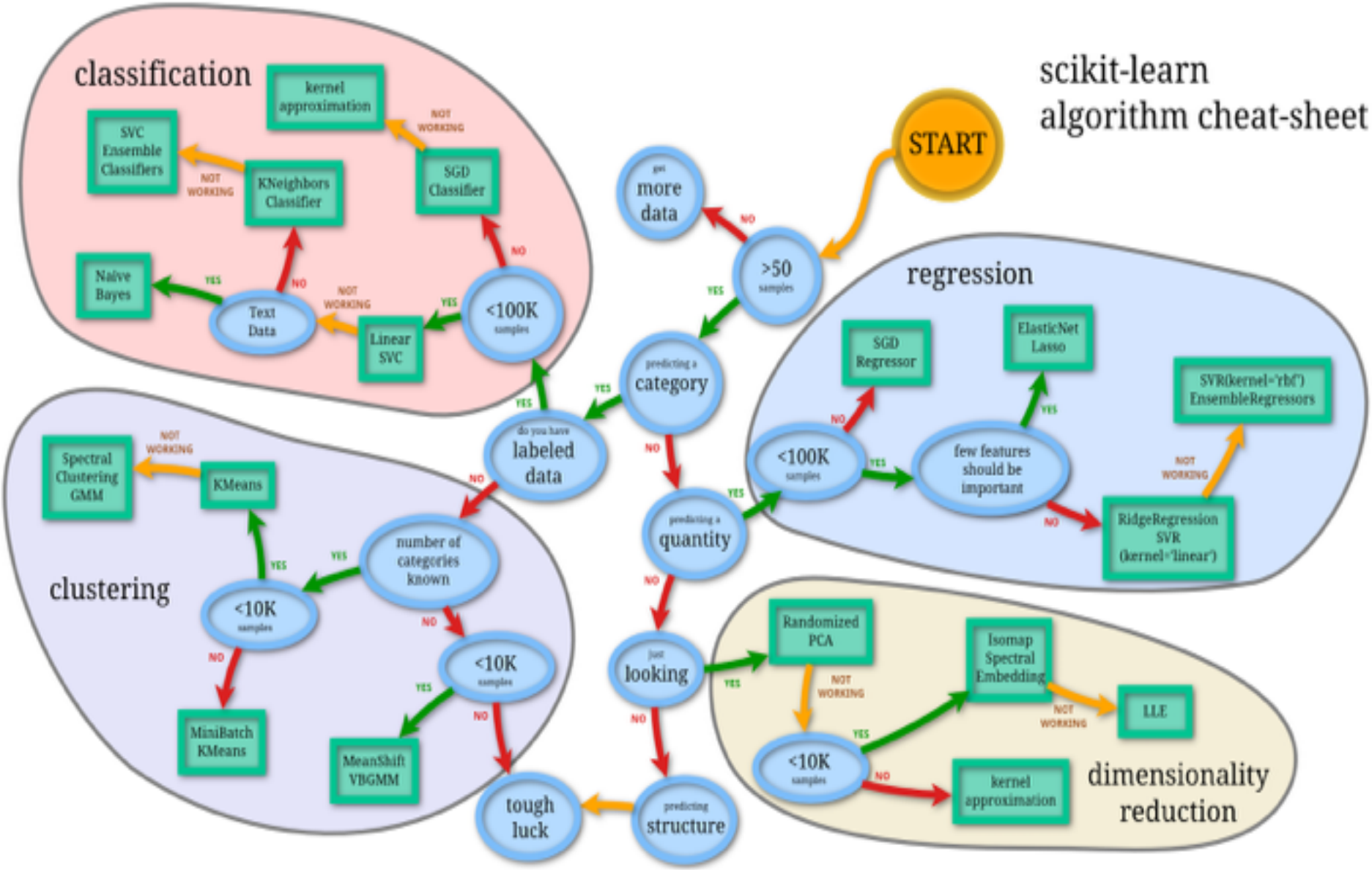
Statsmodel

- **Linear regression models:**
  - Generalized least squares (including weighted least squares and least squares with autoregressive errors),
  - ordinary least squares.
  - glm: Generalized linear models with support for all of the one-parameter exponential family distributions.
  - discrete: regression with discrete dependent variables, including Logit, Probit, MNLogit, Poisson, based on maximum likelihood estimators
  - rlm: Robust linear models with support for several M-estimators.
  - tsa: models for time series analysis - univariate time series analysis:
  - AR, ARIMA - vector autoregressive models,
  - VAR and structural VAR - descriptive statistics and process models for time series analysis nonparametric : (Univariate) kernel density estimators

Scikit
Generalized Linear Models

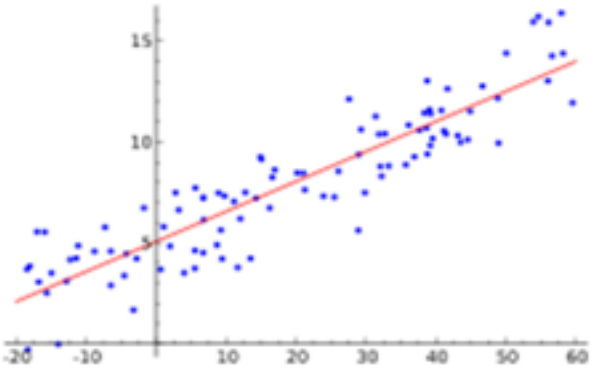
http://scikit-learn.org/stable/modules/linear_model.html

- **Ordinary Least Square**
- **Ridge Regression**
- **Lasso**
- **Elastic Net**
- **Least Angel Regression**
- **Bayesian Regression**
- **Logistic Regression**
- **Stochastic Gradient Decent**


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
YES
Linear SVC
<100K samples
YES
get more data
NO
>50 samples
YES
regression
SGD Regressor
ElasticNet Lasso
SVR(kernel='rbf') EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
NO
RidgeRegression SVR (kernel='linear')
predicting a category
YES
NO
do you have labeled data
YES
NO
clustering
Spectral Clustering GMM
NOT WORKING
KMeans
YES
number of categories known
YES
<10K samples
NO
NO
<10K samples
YES
NO
MiniBatch KMeans
MeanShift VBGMM
predicting a quantity
YES
NO
just looking
YES
Randomized PCA
NOT WORKING
Isomap Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck

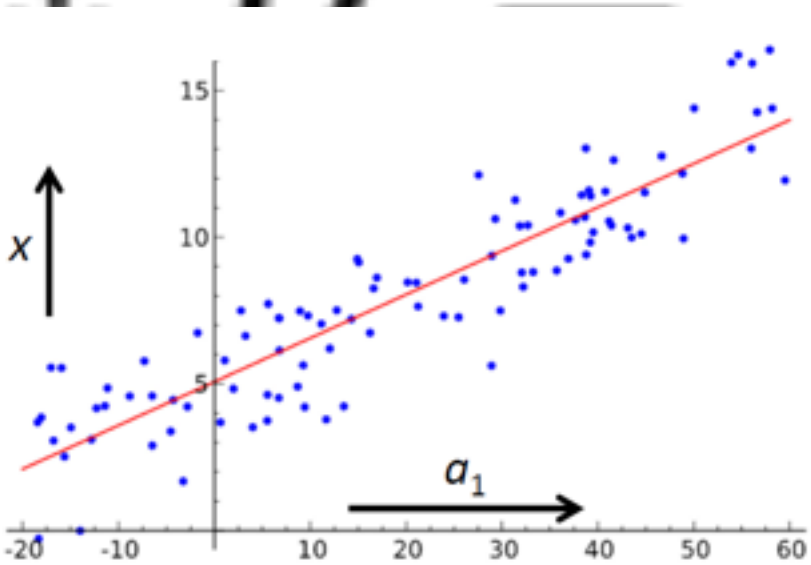
Linear Regression: Description

- **Linear regression is a prediction technique used when the class and all attributes are numeric**
- **One of the easiest technique to use**
- **Bound by "linearity"**
  - If data exhibits a linear dependency, the best-fitting straight line will be found, where best is interpreted as the least mean-squared difference.



Linear Models: Linear Regression

- $y = \sum_{j=0}^{1} a_j x_j$

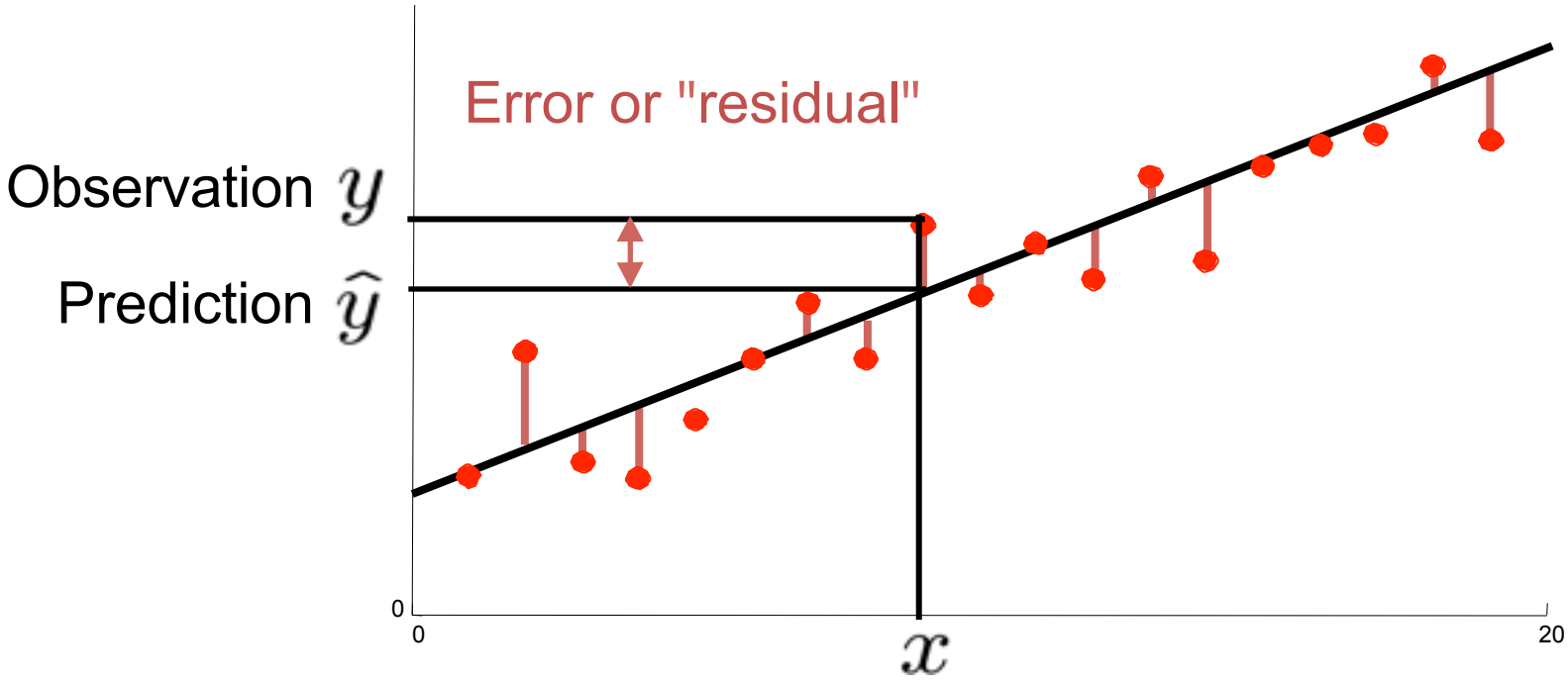


Linear Models: Linear Regression

- $$X * B = -1$$

Ordinary Least Squares (OLS)

$$\text{total error} = \sum_i (y_i - \hat{y}_i)^2 = \sum_i \left(y_i - \sum_k w_k f_k(x_i) \right)^2$$



Linear Regression for House Data

House Size	Lot Size	Bedrooms	Granite	Upgraded	Selling
3529	9191	6	0	0	205,000
3247	10061	5	1	1	224,900
4032	10150	5	0	0	197,900
2397	14156	4	1	0	189900
2200	9600	4	0	1	195000
3536	19994	6	1	1	325000
2983	9365	5	0	1	23000
3198	9669	5	1	1	?????

Linear regression for the Housing data

- **Selling Price =**

  -26.6882 * House Size +

  7.0551 * Lot Size +

  43166.0767 * Bedrooms +

  42292.0901 * Upgraded Bathroom +

  - 21661.1208

Linear regression for the Housing data

- **Selling Price =**

  -26.6882 * 1871+

  7.0551 * 5884 +

  43166.0767 * 4+

  42292.0901 * 1+

  - 21661.1208

Selling price = $184,873.86

Zillow = $448,545.00

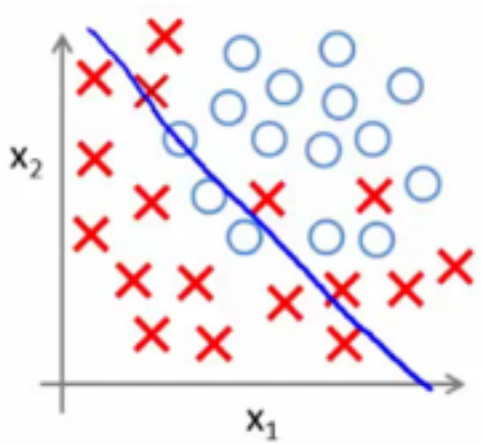
Regression Analysis Assumptions

- The sample is representative of the population for the inference prediction.
- The error is a random variable with a mean of zero conditional on the explanatory variables
- The independent variables are measured with no error
- The predictors are linearly independent, i.e. it is not possible to express any predictor as a linear combination of the others
- The errors are uncorrelated, that is, the variance–covariance matrix of the errors is diagonal and each non-zero element is the variance of the error

OLS Weaknesses

- **Coefficient estimates for OLS rely on the independence of the model terms**
- **When terms are correlated and the columns of the design matrix have an approximate linear dependence - estimate becomes highly sensitive to random errors - producing a large variance**
- ***Multicollinearity* problem**

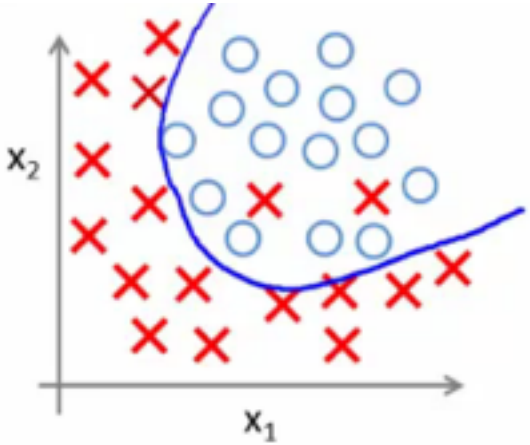
Overfitting



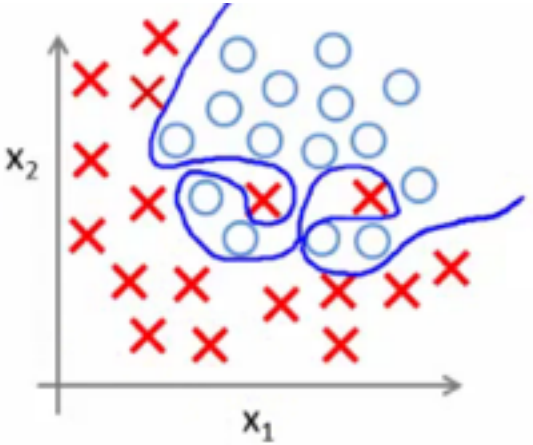
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

UNDERFITTING
(high bias)



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

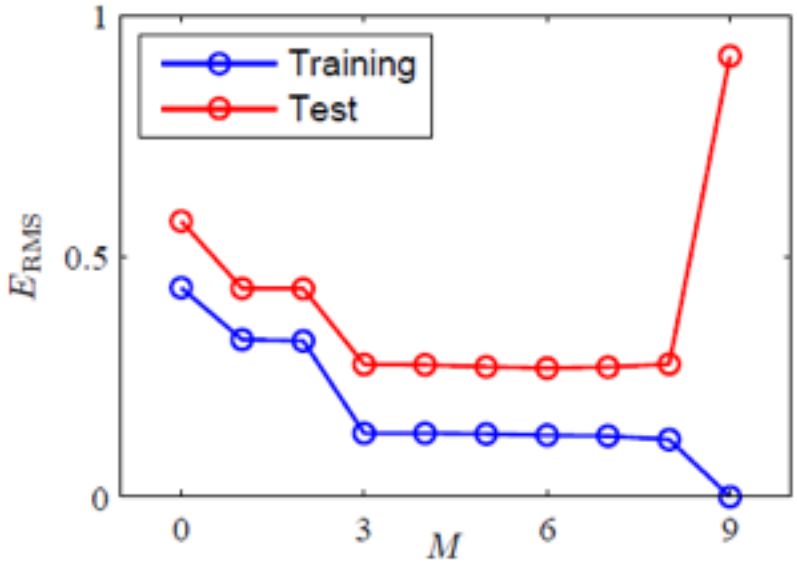


$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$

OVERFITTING
(high variance)

Overfitting Example

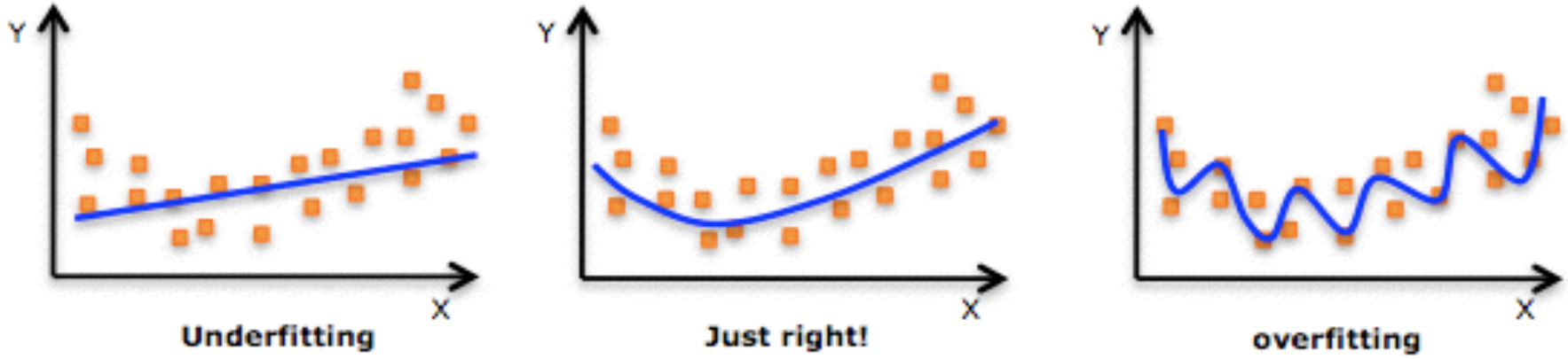
Example from Bishop, Figure 1.5



For any given N, some h of sufficient complexity fits the data but may have very bad generalization error!!

Regularization

Tuning or selecting the preferred level of model complexity so your models are better at predicting (generalizing)



Regularization Concept

- **Non-negative loss function**

  L(actual value, predicted value)

- **Fit your model in a such way that its predictions minimize mean of loss function, calculated only on training data**

  $\text{Model} = \text{argmin} \sum L(\text{actual}, \text{predicted}(\text{Model}))$

- **Explain patterns but also explains random noise**
- **Degradation of generalization ability**
- $\text{Model} = \text{argmin} \sum L(\text{actual}, \text{predicted}(\text{Model})) + \lambda R(\text{Model})$

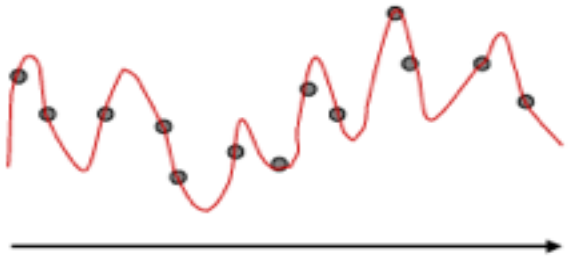
Regularization

– The minimization

$$\min_f |Y_i - f(X_i)|^2$$

may be attained with zero errors.

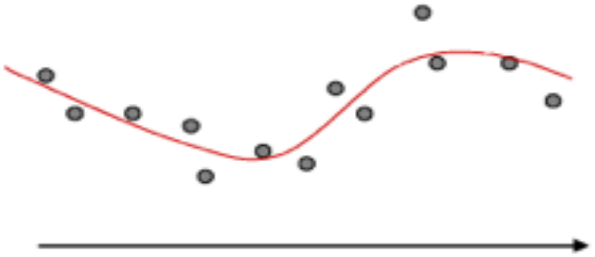
But the function may not be unique.



– Regularization

$$\min_{f \in H} \sum_{i=1}^{n} |Y_i - f(X_i)|^2 + \lambda \|f\|_H^2$$

- Regularization with smoothness penalty is preferred for uniqueness and smoothness.
- Link with some RKHS norm and smoothness is discussed in Sec. IV.



Ridge Regression

- Technique for analyzing multiple regression data that suffer from multicollinearity
- Addresses some of the problems of OLS by imposing a penalty on the size of coefficients
- The ridge coefficients minimize a penalized residual sum of squares

$$\min_{w} ||Xw - y||_2^2 + \alpha ||w||_2^2$$

- Complexity parameter that controls the amount of shrinkage: the larger the value of , the greater the amount of shrinkage and thus the coefficients become more robust to collinearity

Lasso

- Least absolute shrinkage and selection operator
- Linear model that estimates sparse coefficients
- Tendency to prefer solutions with fewer parameter values
- Reducing the number of variables upon which the given solution is dependent

$$\min_{w} \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha ||w||_1$$

Elastic Net

- **Overcomes the limitations of Lasso**
- **Large *p*, small *n*" case - high-dimensional data with few examples or group of highly correlated variables**
- **Adds a quadratic part to the penalty** $\|\beta\|^2$

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1).$$

- **includes the LASSO and ridge regression**

Logistic regression

- **Analyze relationships between a dichotomous dependent variable and metric or dichotomous independent variables**
- **The log odds of the outcome is modeled as a linear combination of the predictor variables**
- **Combines the independent variables to estimate the probability that a particular event will occur, i.e. a subject will be a member of one of the groups defined by the dichotomous dependent variable**
- **Classification using Regression?**

What Logistic Regression Predicts

- The variate or value produced by logistic regression is a probability value between 0.0-1.0
- Probability for group membership in the modeled category is above/below some cut point (the default is 0.50) determines the group
- For any given case, logistic regression computes the probability that a case with a particular set of values for the independent variable is a member of the modeled category

Logistic Regression

- **Models relationship between set of variables X_i**
  - dichotomous (yes/no, smoker/nonsmoker)
  - categorical (social class, race)
  - continuous (age, weight, gestational age)

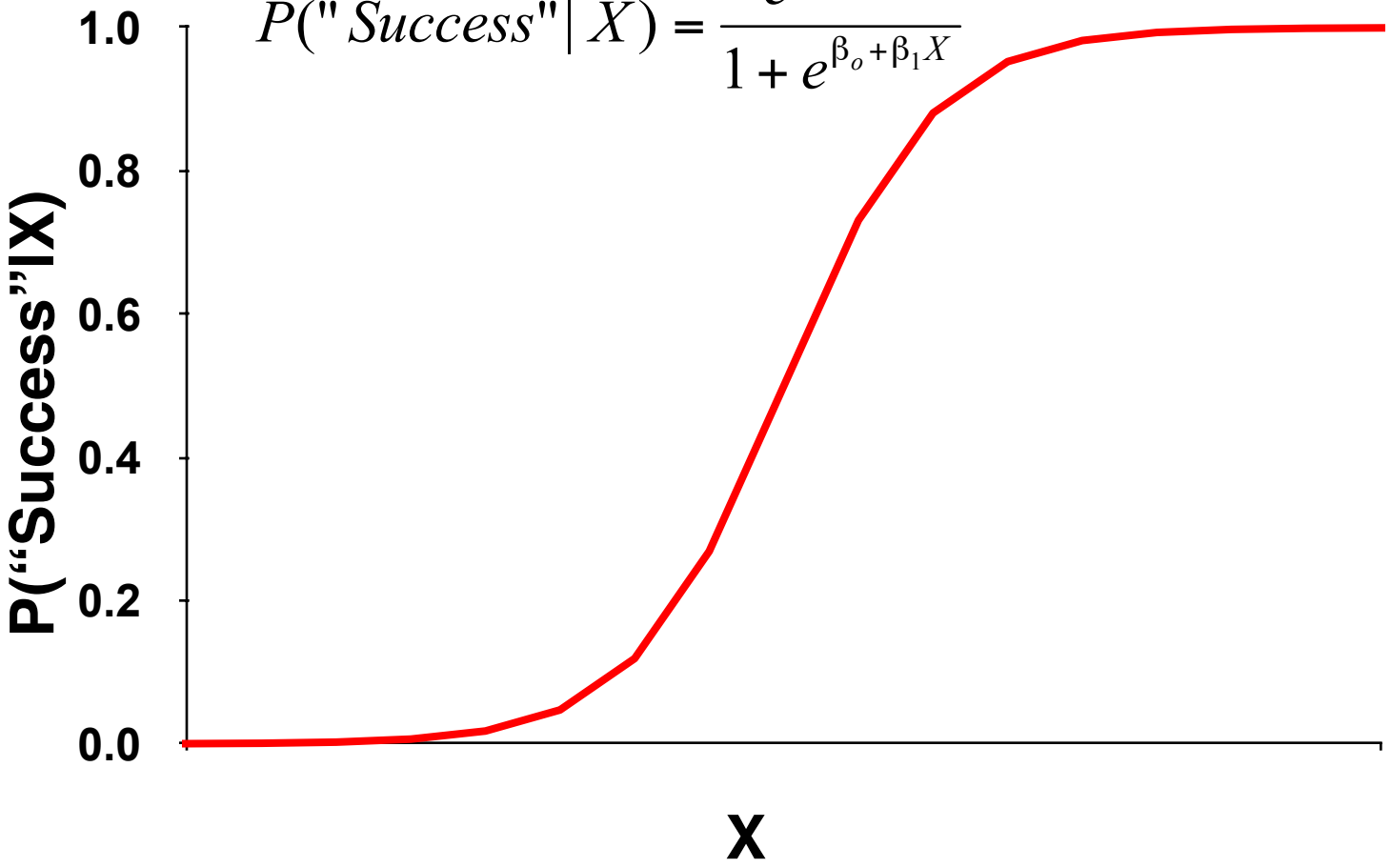
and

  - dichotomous categorical response variable *Y*

    e.g. Success/Failure, Remission/No Remission Survived/Died, CHD/No CHD, Low Birth Weight/Normal Birth Weight

Logistic Function

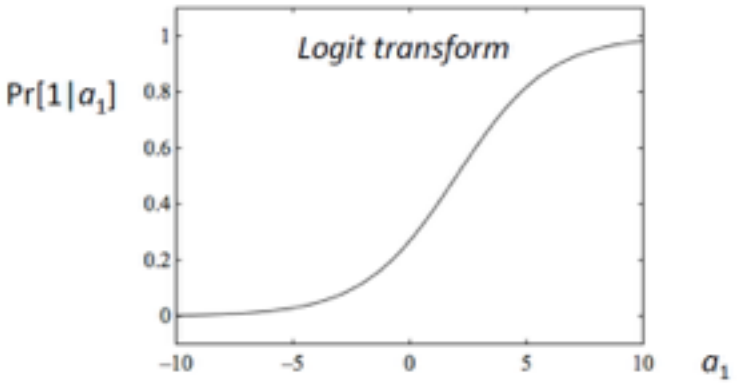
$$P(\text{"Success"} \mid X) = \frac{e^{\beta_o + \beta_1 X}}{1 + e^{\beta_o + \beta_1 X}}$$



Logit Transform

- **Linear regression calculate a linear function and threshold**
- **Logistic Regression estimate class probabilities directly**

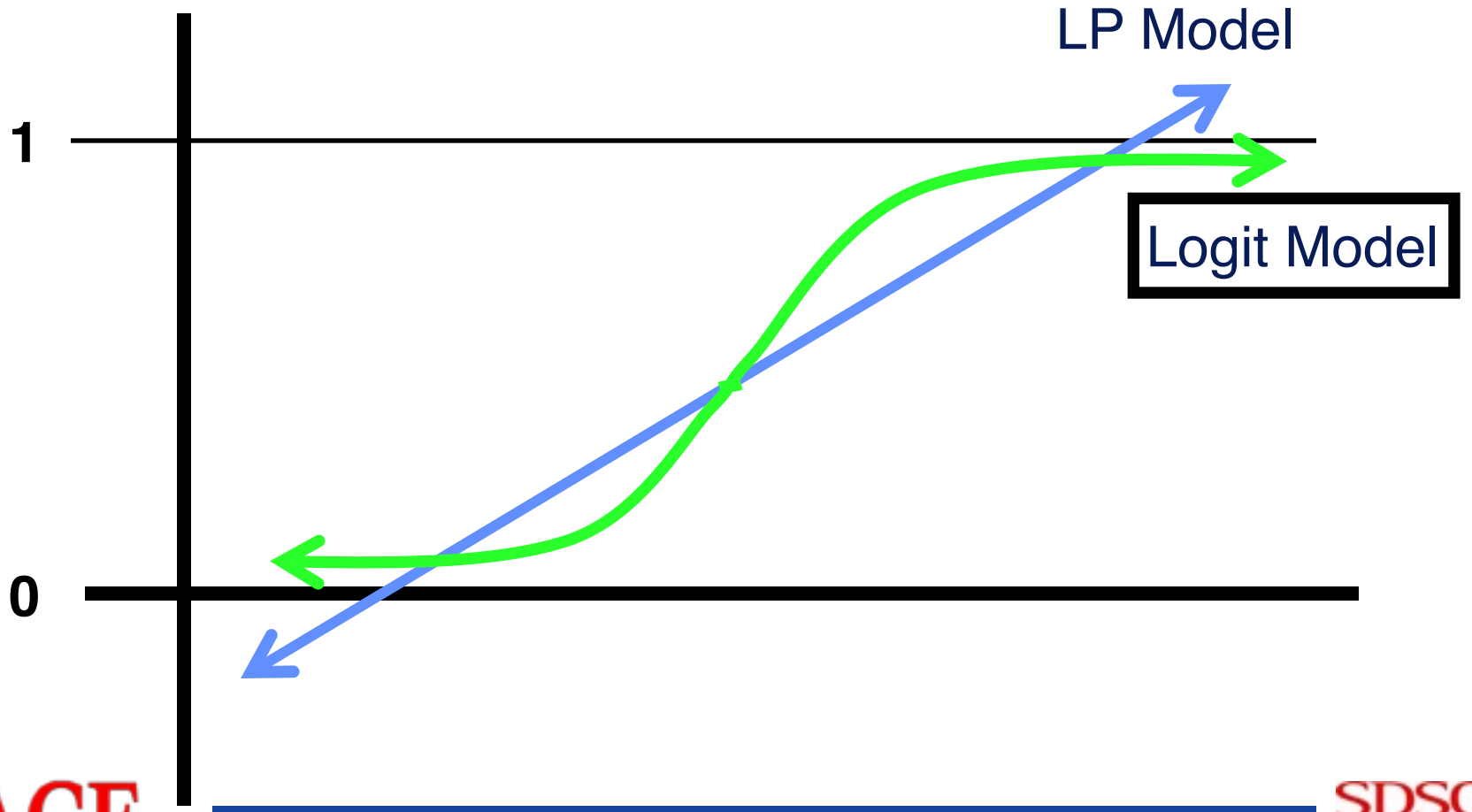
$$\Pr[1 \mid a_1, a_2, \ldots, a_k] = 1/(1 + \exp(-w_0 - w_1 a_1 - \ldots - w_k a_k))$$



- **Choose the weights to maximize the log-likelihood**

$$\sum_{i=1}^{n} (1 - x^{(i)}) \log(1 - \Pr[1 \mid a_1^{(1)}, a_2^{(2)}, \ldots, a_k^{(k)}]) + x^{(i)} \log(\Pr[1 \mid a_1^{(1)}, a_2^{(2)}, \ldots, a_k^{(k)}])$$

Comparing LP and Logit Models



When To Use Logistic Regression

- In logistic regression the response (Y) is a dichotomous categorical variable
- Uses logit transform to predict probabilities directly (similar to Naïve Bayes)
- Logit is popular and powerful
- Used widely in many fields, including the medical and social sciences
- Predict whether an American voter will vote Democratic or Republican

Generalized Models in Scikit

- **http://scikit-learn.org/stable/modules/linear_model.html**

Linear regression equation for the CPU data

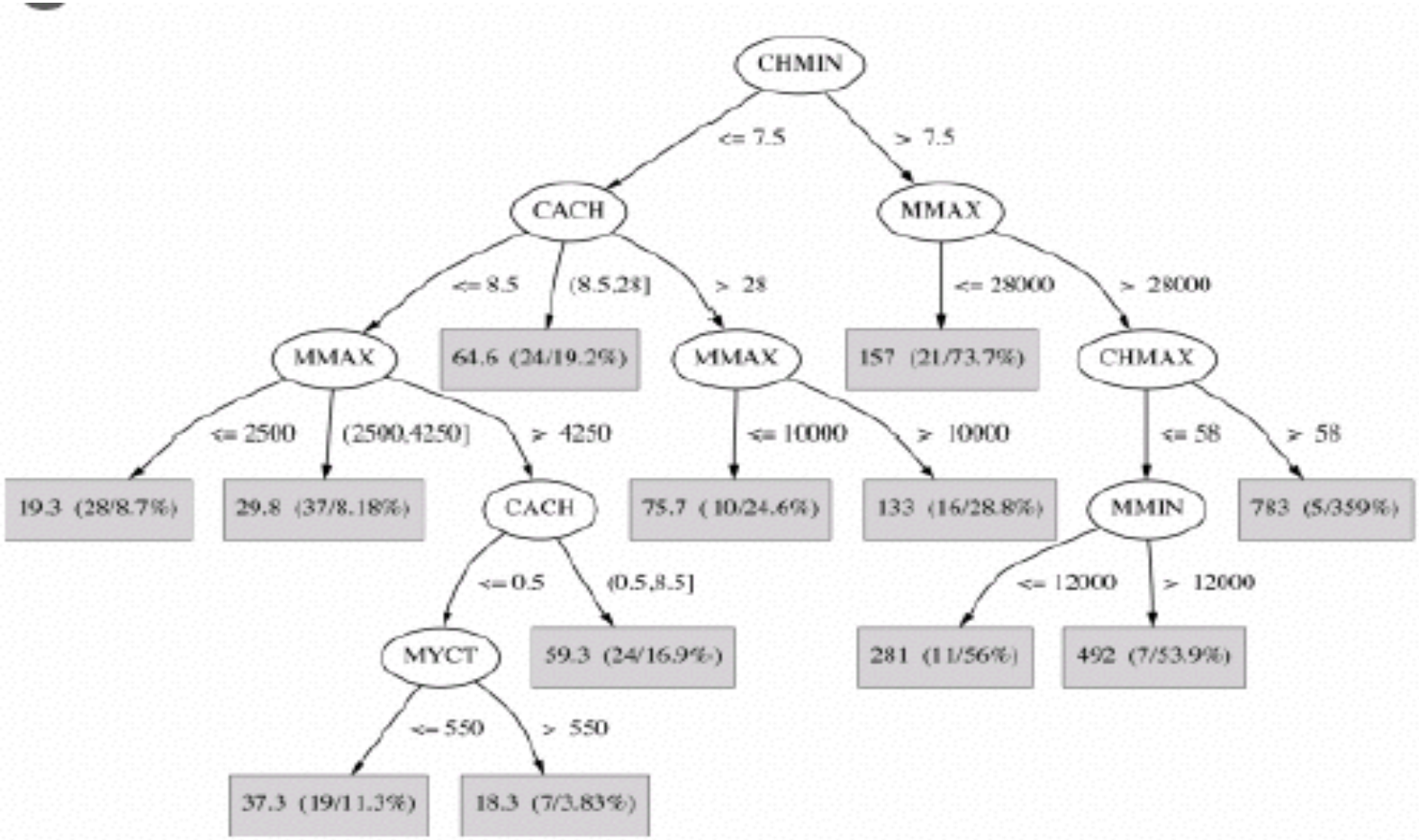
- **PRP =**
  - **- 56.1**
  - **+ 0.049 MYCT**
  - **+ 0.015 MMIN**
  - **+ 0.006 MMAX**
  - **+ 0.630 CACH**
  - **- 0.270 CHMIN**
  - **+ 1.46 CHMAX**

	Cycle Time	Main Memory (Kb)		Cache (Kb)	Channels		Performance
	MYCT	*MMIN*	*MMAX*	*CACH*	*CHMIN*	*CHMAX*	*PRP*

Non-linear Regression: Regression and Model trees

- **Similar to decision trees**
- **Modifications**
  - Splitting criterion to minimize intra-subset variation
  - Termination criterion reached when standard deviation becomes insignificant
  - Pruning criterion based on numeric error measure
  - Leaf predicts average class values of instances
- **Easy to interpret**
- **Python: http://scikitlearn.org/0.11/modules/tree.html#classification**

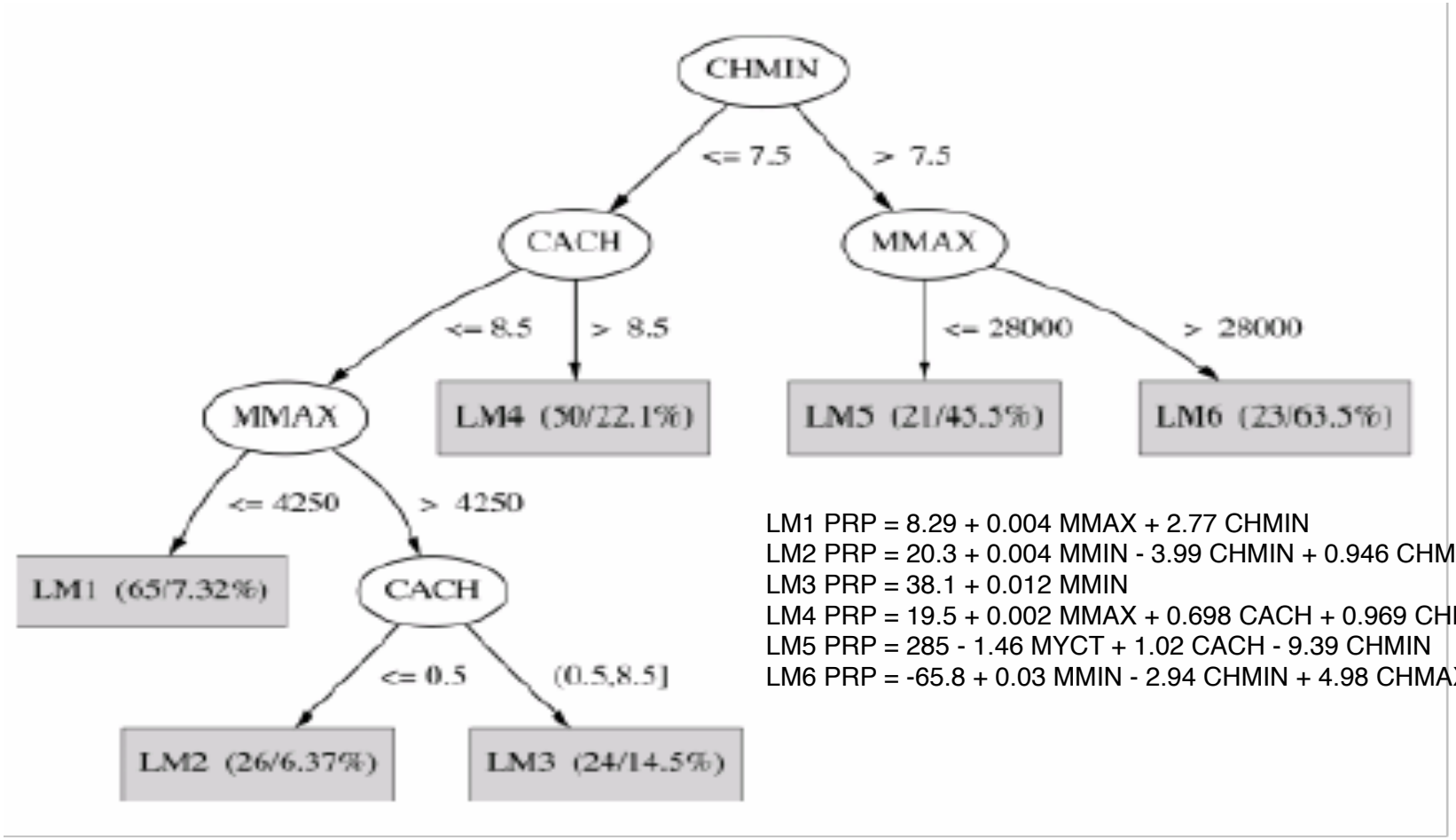
Regression tree for the CPU data



Model trees

- **Build a regression tree**
- **Each leaf -> linear regression function**
- **Smoothing: factor in ancestor's predictions**
  - Smoothing formula: (function) p' = (np+kq)/(n+k)
  - Same effect can be achieved by incorporating ancestor models into the leaves
- **Need linear regression function at each node**
- **At each node, use only a subset of attributes**
  - Those occurring in subtree (+maybe those occurring in path to the root)
- **Fast: tree usually uses only a small subset of the attributes**

Model tree for CPU data set



LM1 PRP = 8.29 + 0.004 MMAX + 2.77 CHMIN
LM2 PRP = 20.3 + 0.004 MMIN - 3.99 CHMIN + 0.946 CHMAX
LM3 PRP = 38.1 + 0.012 MMIN
LM4 PRP = 19.5 + 0.002 MMAX + 0.698 CACH + 0.969 CHMAX
LM5 PRP = 285 - 1.46 MYCT + 1.02 CACH - 9.39 CHMIN
LM6 PRP = -65.8 + 0.03 MMIN - 2.94 CHMIN + 4.98 CHMAX

Model Trees: Building the tree (Splitting)

- **Splitting: standard deviation reduction**
- **SDR=sd(T)–Σi |Ti/T |×sd(Ti) (function)**
- **Termination:**
  - Small Standard Deviation (example < 5% of its value on full training set
  - Too few instances remain (e.g. < 4)

Model Trees: Building a Tree (Pruning)

- **Heuristic estimate of absolute error of LR models:**
- **Function (n+v)/(n-v) * average_absolute_error**
- **Greedily remove terms from LR models to minimize estimated error**
- **Heavy pruning: single model may replace whole subtree**
- **Proceed bottom up: compare error of LR model at internal node to error of subtree**

Evaluating Numeric Prediction

- **Same strategies: independent test set, cross validation, significance tests, etc.**
- **Difference: error measures**
- **Most popular measure: mean squared error**
- **Easy to manipulate mathematically**

Summary

- **Regression: the process of computing an expression that predicts a numeric quantity**
- **Linear Regression: simple method for numeric prediction of for linear data**
- **Regression Tree: "decision tree" where each leaf predicts a numeric quantity**
  - Predicted value is average value of training instances that reach the leaf
- **Model Tree : "regression tree" with linear regression models at the leaf nodes**
  - Linear patches approximate continuous function
    - Linear regression model that predicts the class value of instances that reach the leaf