iSmartsoft

Comparing Different Classification Techniques in Credit Scoring

Saed Sayad

Data Mining

Data mining is about explaining the past and predicting the future by means of data analysis

Credit Scoring

Credit scoring is a statistical method that is used to predict the probability that a loan applicant or existing borrower will default or become delinquent

(Mester, 1997)

Data Mining in Credit Scoring

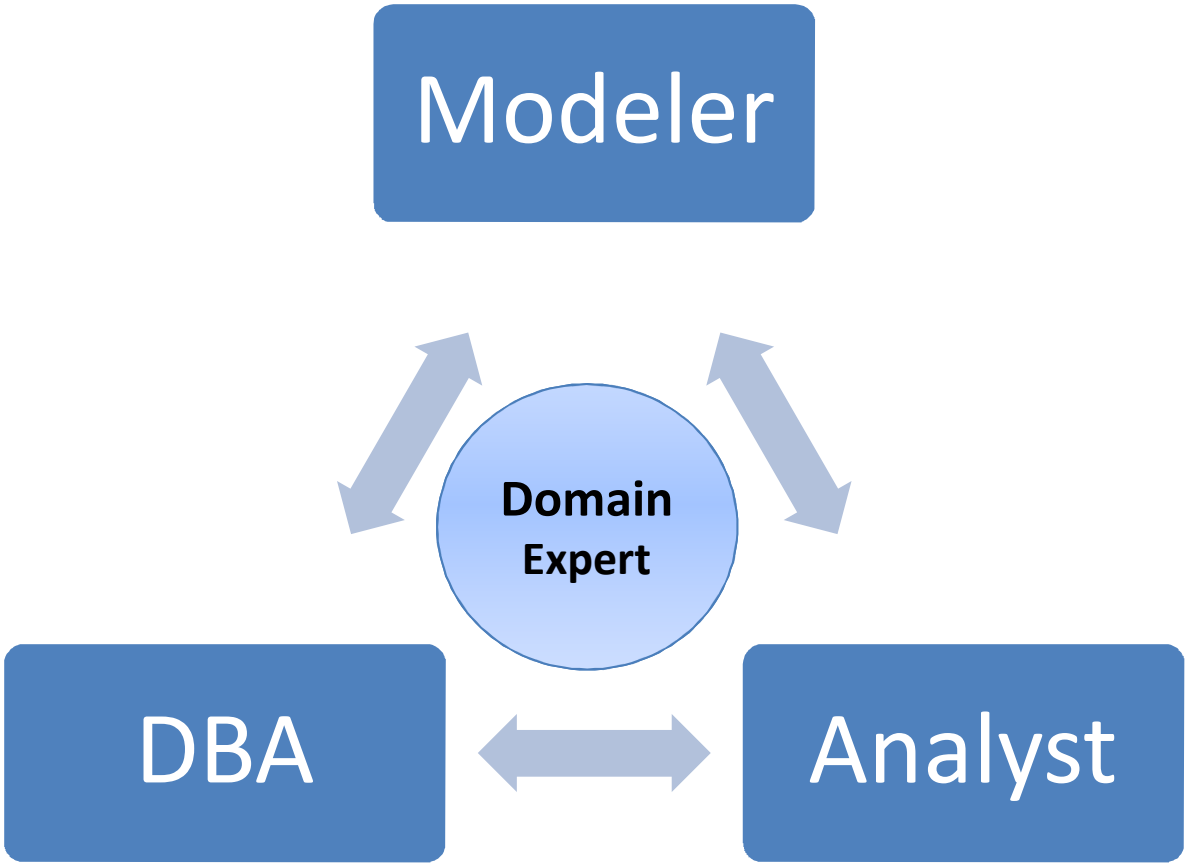
1. Problem Definition
2. Data Preparation
3. Data Exploration
4. Modeling
5. Evaluation
6. Deployment

1. Problem Definition

- Develop a credit scoring model to predict the credit risk of credit applicants as bad risk (default) and good risk.

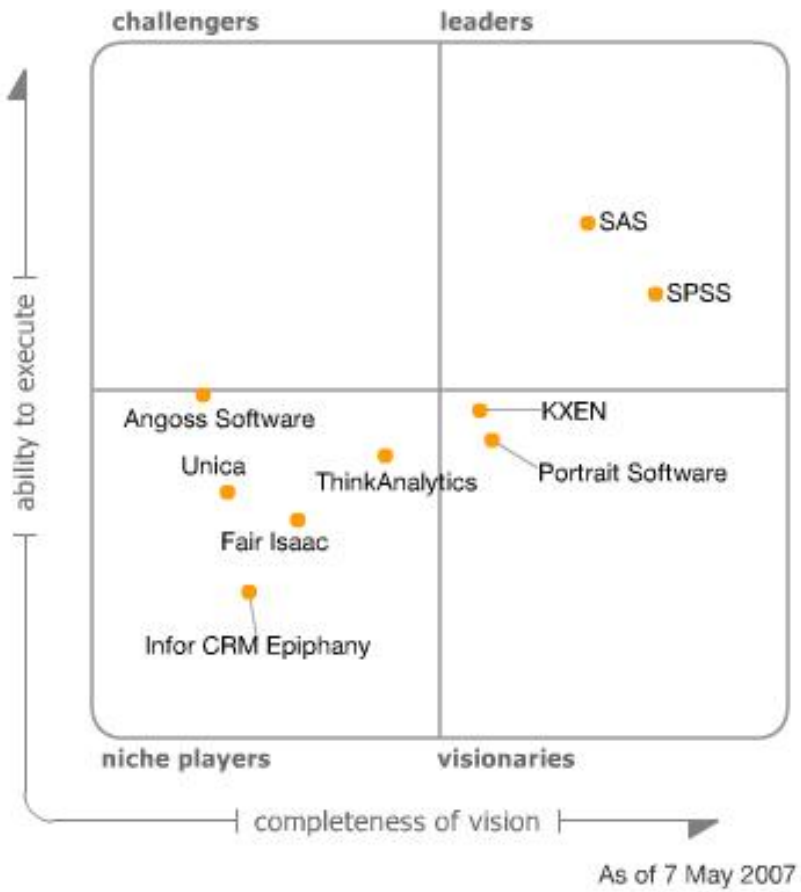
- The credit issuer intends to deploy the model for the *existing* borrower.

Data Mining Team



Data Mining Software Vendors

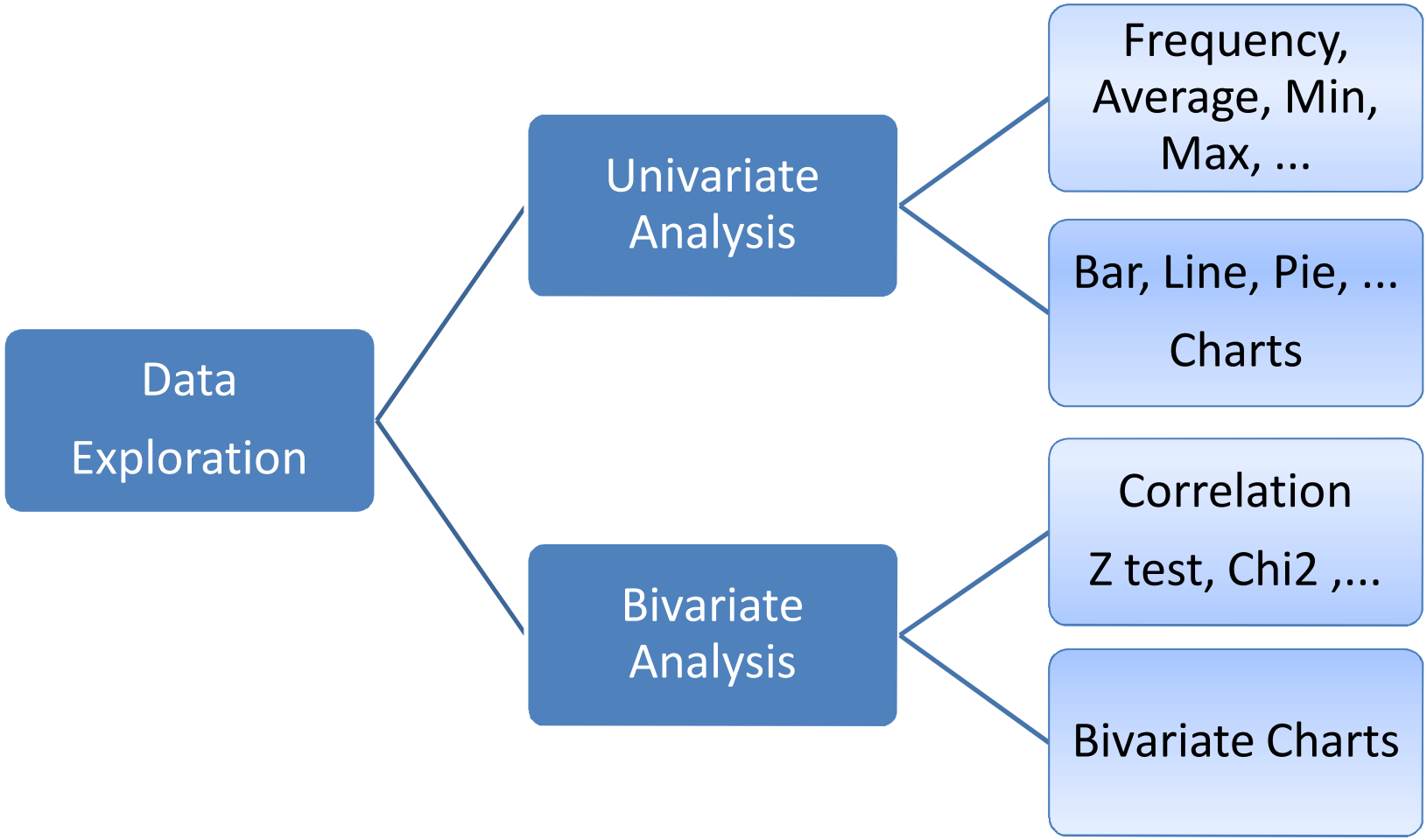
Gartner Magic Quadrant



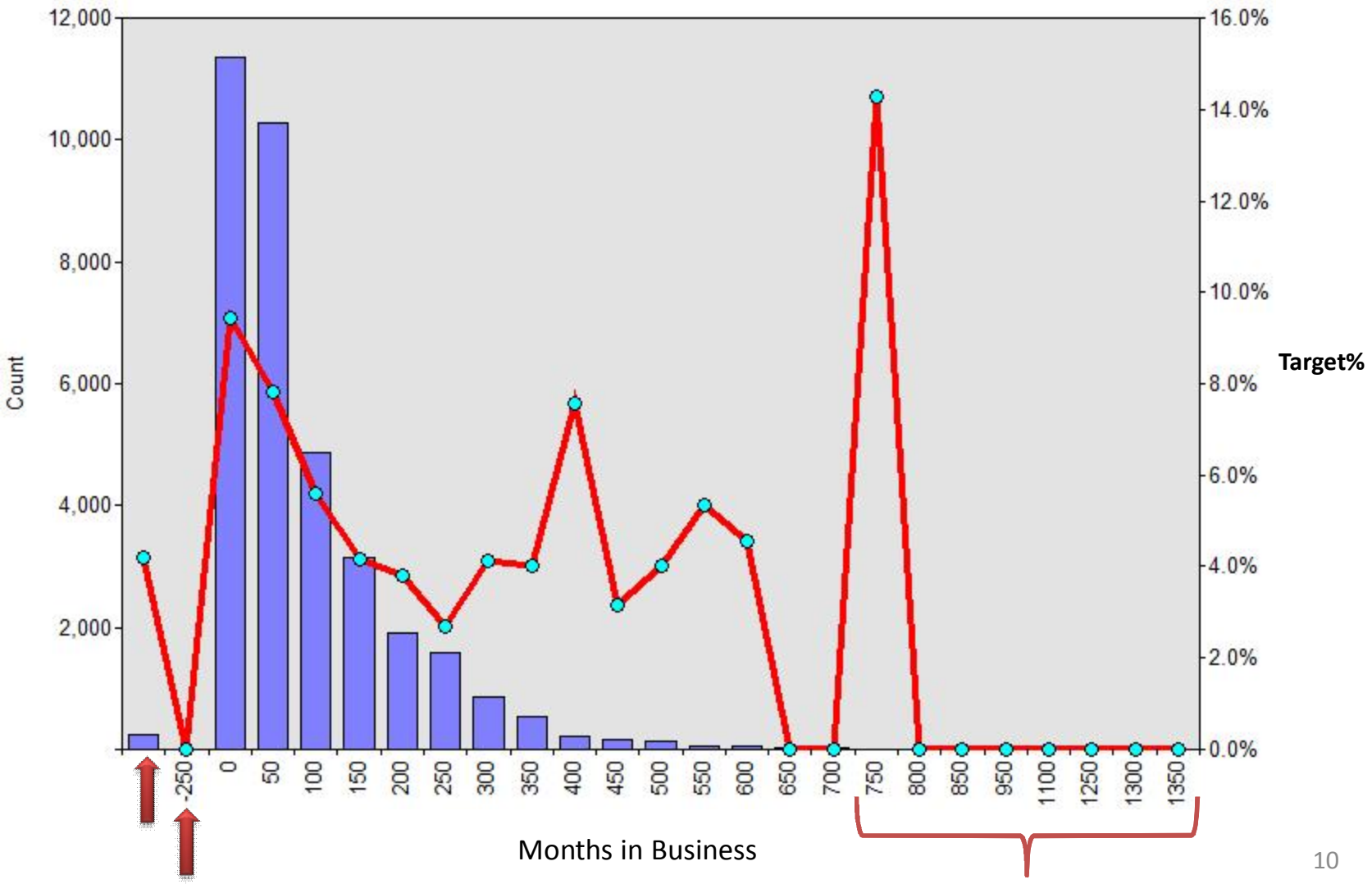
2. Data Preparation

- **Training set**
  - No of cases: **35,500**
  - No of cases with bad risk (target): **2,500 (7%)**
  - Number of variables: **25**
  - Total balance for all cases: **$554,000,000**
  - Total balance for cases with bad risk: **$58,000,000**
- **Test set**
  - Total number of cases: **8,167**
  - Total number of targets: **560**
  - Total balance for cases with bad risk : **$12,281,589**

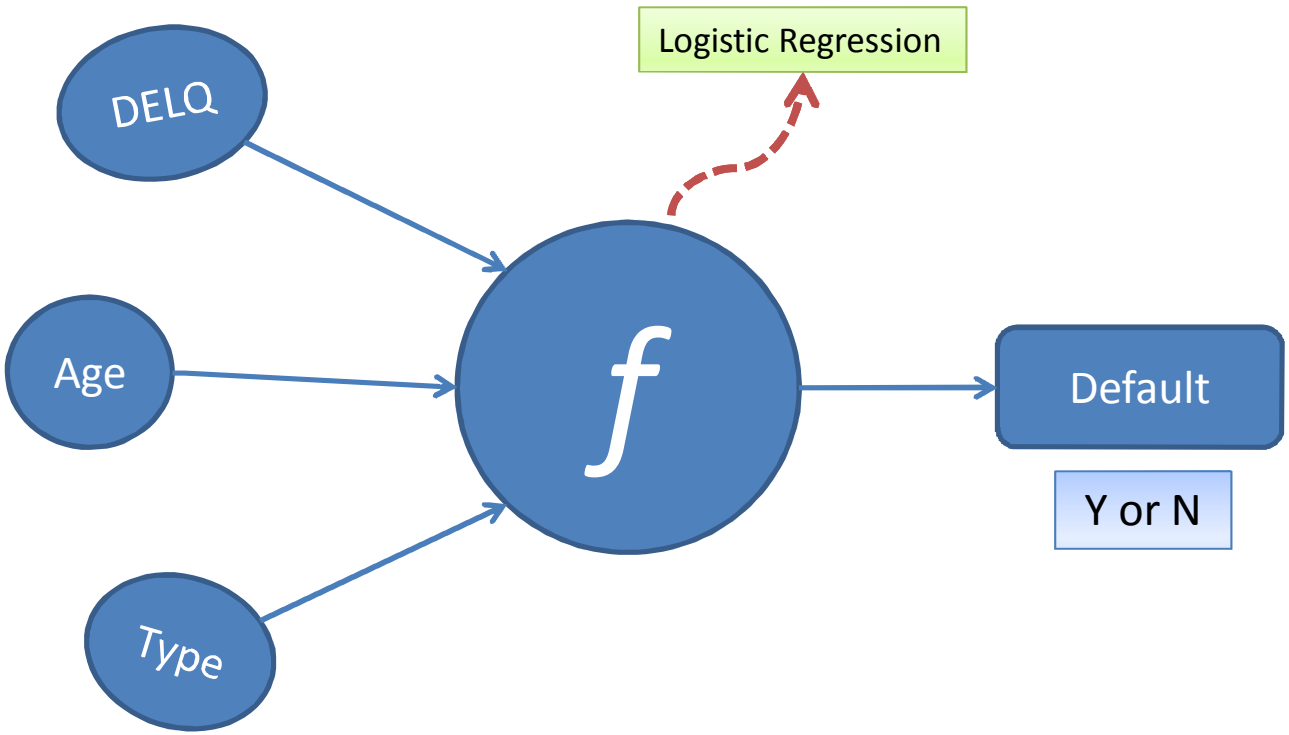
3. Data Exploration



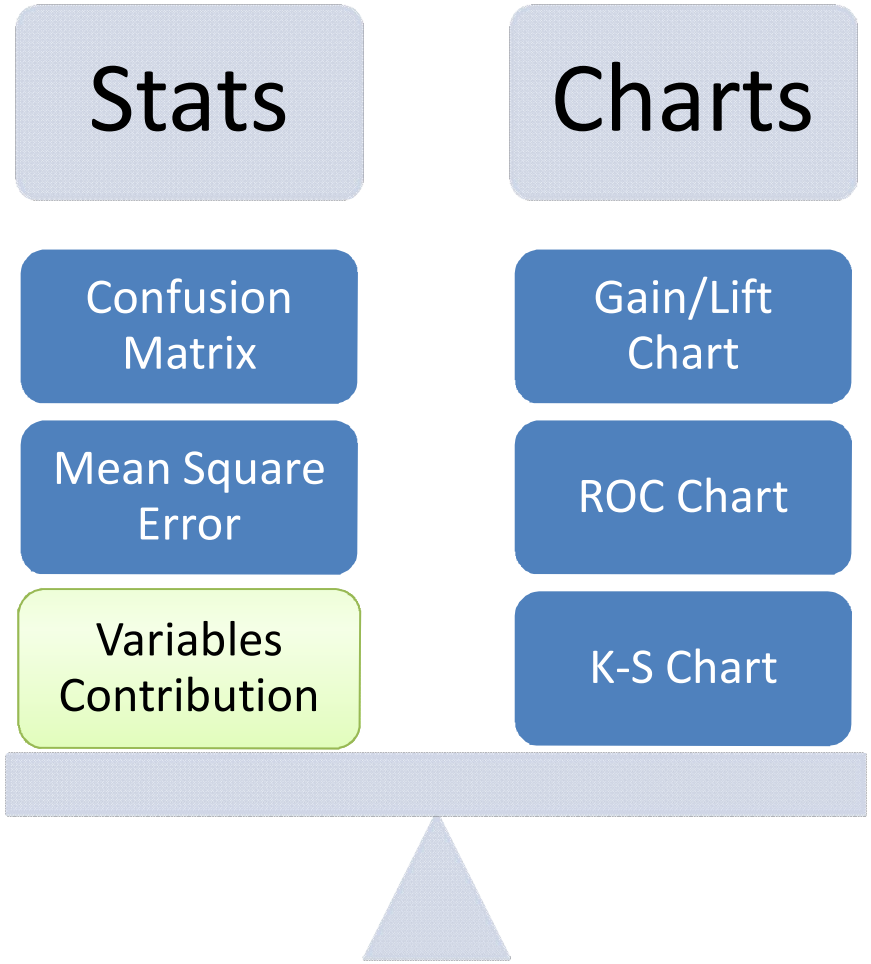
Data Exploration - Bivariate



4. Modeling



5. Evaluation

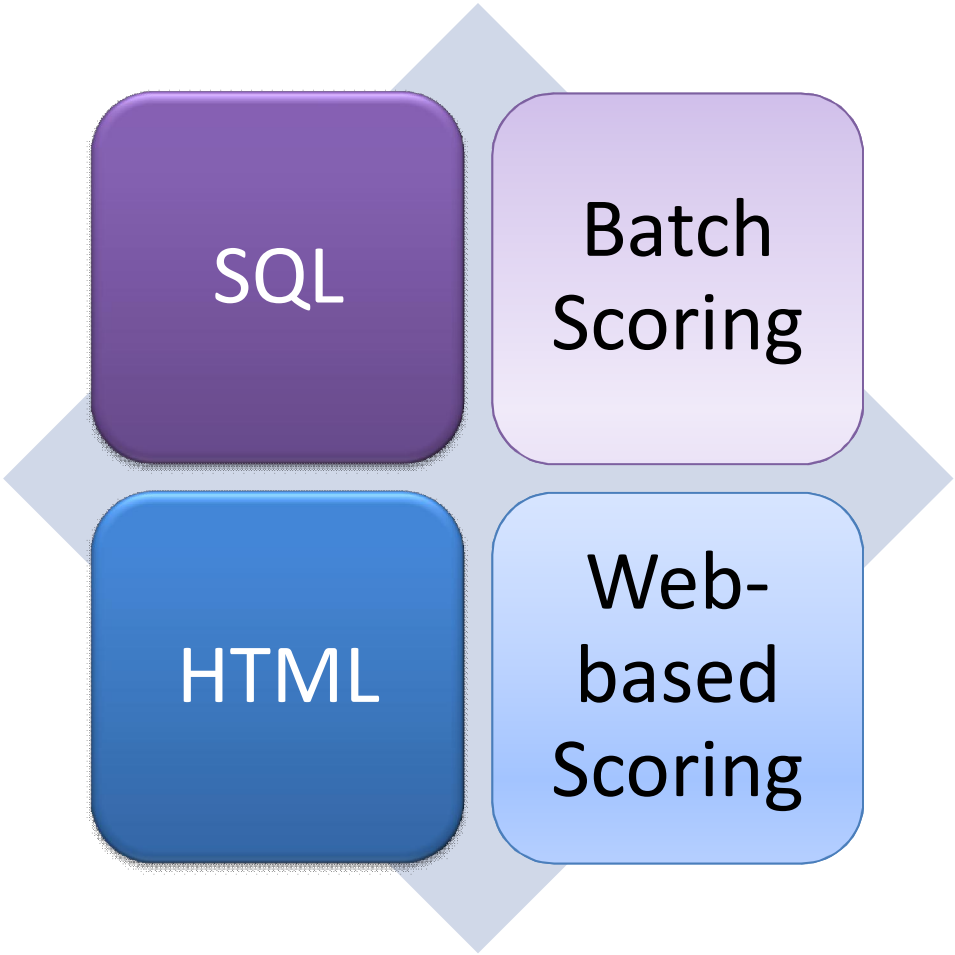


Result

- Total number of cases = **8,167**
- Total number of targets = **560**
- Total balance for targets = **$12,281,589**
- Top 10% **Random**
  - Number of targets = **56**
  - Total balance = **$1,230,000**
- Top 10% **Model**
  - Number of targets = **305**
  - Total balance = **$7,655,772**

600% Lift

6. Deployment

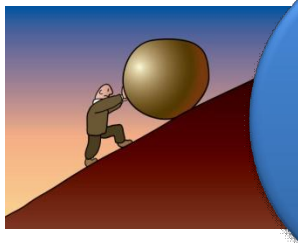


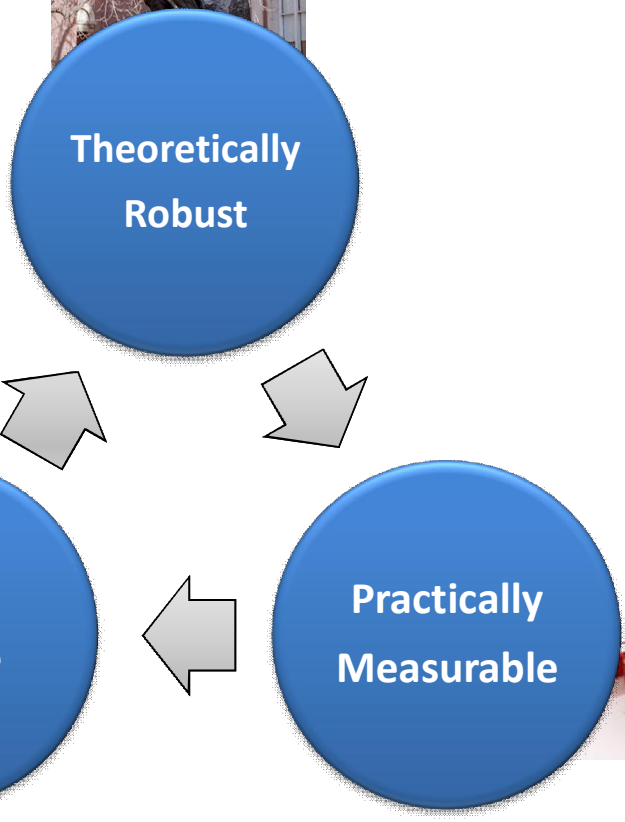
Do we need to try other modeling techniques?

If so, how do we compare them?

Model Comparison

No black box please!



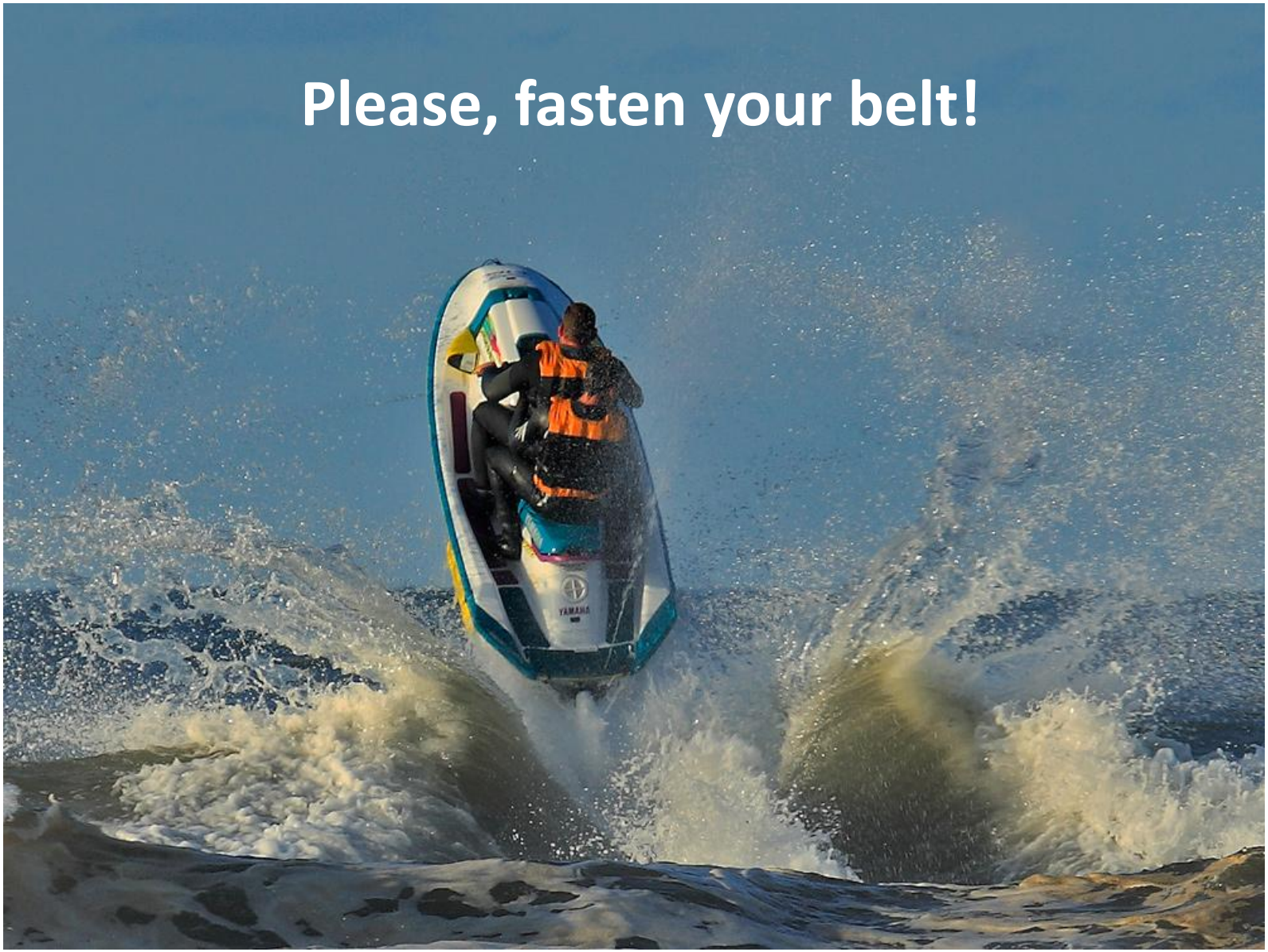


Don’t be Sisyphus of the analytics world!

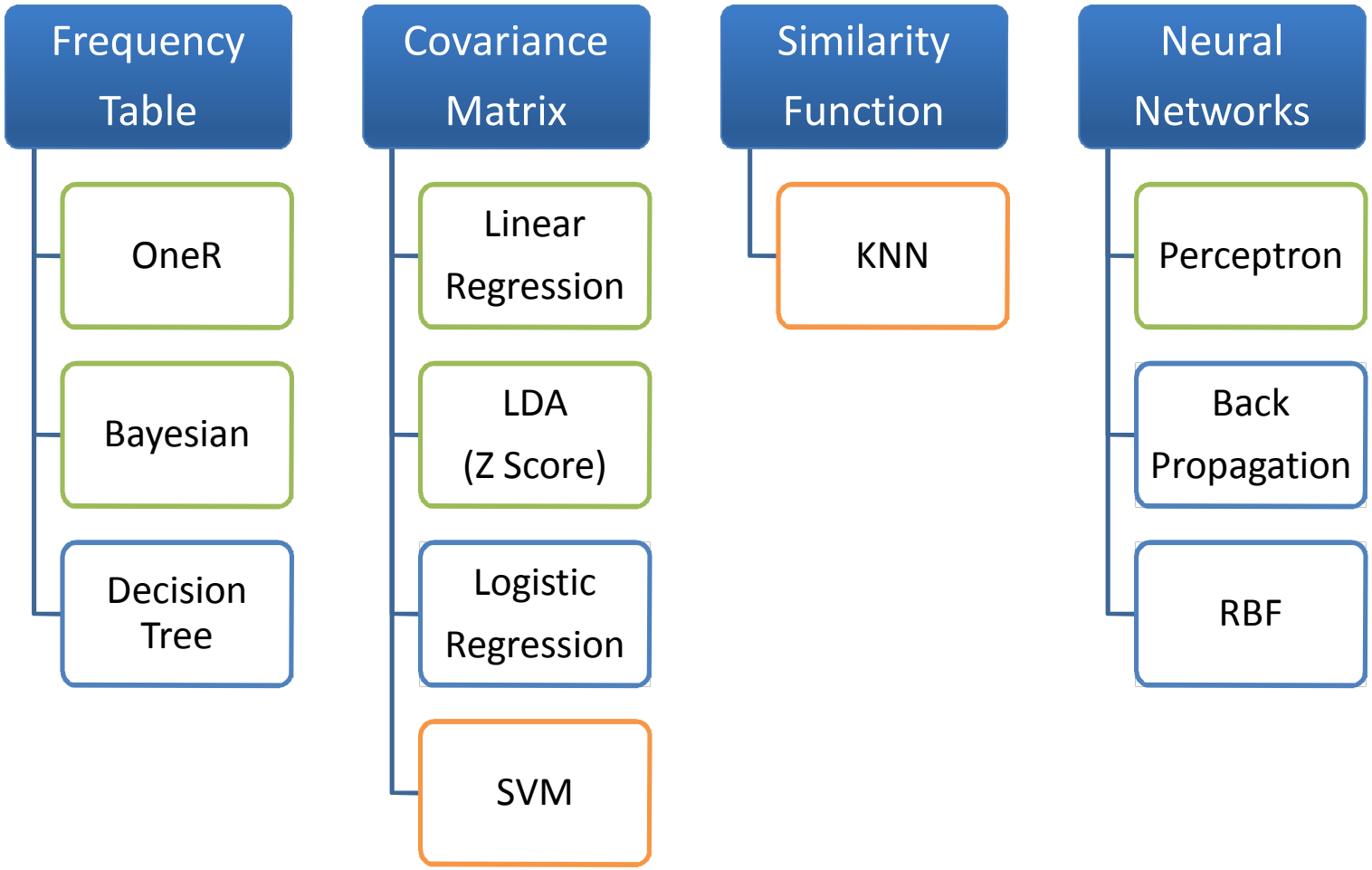
Inputs & Outputs

True Scalability

- **Incremental/Decremental Learning**: utilizing new data without the necessity of pooling new data with old data and repeating the model training step;
- **Variable Addition/Deletion**: add or remove variables without re-training;
- **Scenario Testing:** rapid formulation and testing of multiple and diverse models;
- **Parallel and Distributed Processing:** carrying out parallel and/or distributed processing simultaneously for all datasets or data segments to enable a single model using all of the data to be obtained.


Please, fasten your belt!

Modeling (Classification & Regression)



Scalable Methods

Models
based on
Frequency Tables

ZeroR Classification

Predictors

Target

Play Golf
No
No
Yes
Yes
Yes
No
Yes
No
Yes
Yes
Yes
Yes
Yes
No

Classification - ZeroR

Play Golf
No
No
Yes
Yes
Yes
No
Yes
No
Yes
Yes
Yes
Yes
No

Sort

Play Golf
No
No
No
No
No
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes

5 / 14 = 0.36

9 / 14 = 0.64

Yes

Classification - ZeroR

Default

2493/35500 = **0.07**

33007/35500 = **0.93**

No

Classification - OneR

Which one is the best predictor ?

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

OneR - Frequency Tables

		Play Golf	
		Yes	No
Outlook	Sunny	**3**	**2**
	Overcast	**4**	**0**
	Rainy	**2**	**3**

		Play Golf	
		Yes	No
Temp.	Hot	**2**	**2**
	Mild	**4**	**2**
	Cool	**3**	**1**

		Play Golf	
		Yes	No
Humidity	High	**3**	**4**
	Normal	**6**	**1**

		Play Golf	
		Yes	No
Windy	False	**6**	**2**
	True	**3**	**3**

OneR – The Best Rule

MAXDELQ:

< 2 -> N

>= 2 -> Y

? -> N

94%

Can we incorporate all predictors?

Bayesian Classifier

Bayes' Rule

Likelihood

Target Prior Probability

$$P(T \mid \mathrm{X}) = \frac{P(\mathrm{X} \mid T)\, P(T)}{P(\mathrm{X})}$$

Posterior Probability

Predictor Prior Probability

Likelihood Tables

		Play Golf	
		Yes	No
Outlook	Sunny	**3/9**	**2/5**
	Overcast	**4/9**	**0/5**
	Rainy	**2/9**	**3/5**

		Play Golf	
		Yes	No
Temp.	Hot	**2/9**	**2/5**
	Mild	**4/9**	**2/5**
	Cool	**3/9**	**1/5**

		Play Golf	
		Yes	No
Humidity	High	**3/9**	**4/5**
	Normal	**6/9**	**1/5**

		Play Golf	
		Yes	No
Windy	False	**6/9**	**2/5**
	True	**3/9**	**3/5**

$$P(Outlook = Sunny \mid PlayGolf = Yes) = 3/9 = 0.33$$

Predictor Probability

		Play Golf		
		Yes	No	
Outlook	Sunny	**3/9**	**2/5**	**5/14**
	Overcast	**4/9**	**0/5**	**4/14**
	Rainy	**2/9**	**3/5**	**5/14**

$$P(Outlook = Sunny) = 5/14 = 0.36$$

Target Probability

		Play Golf		
		Yes	No	
Outlook	Sunny	**3/9**	**2/5**	**5/14**
	Overcast	**4/9**	**0/5**	**4/14**
	Rainy	**2/9**	**3/5**	**5/14**
		9/14	**5/14**	

$$P(PlayGolf = Yes) = 9/14 = 0.64$$

Posterior Probability

$$P(Outlook = Sunny \mid PlayGolf = Yes) = 3/9 = 0.33$$

$$\times$$

$$P(PlayGolf = Yes) = 9/14 = 0.64$$

$$\div$$

$$P(Outlook = Sunny) = 5/14 = 0.36$$

$$\Downarrow$$

$$P(PlayGolf = Yes \mid Outlook = Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$$

$$P(T \mid \mathrm{X}) = \frac{P(\mathrm{X} \mid T)\,P(T)}{P(\mathrm{X})}$$

Bayesian - Prediction

Case = [Outlook=*Sunny*; Humidity=*High*; Temp=*Mild*; Windy=*True*]

Posterior Probability		Play Golf	
		Yes	No
Outlook	Sunny	**0.60**	**0.40**

Posterior Probability		Play Golf	
		Yes	No
Temp.			
	Mild	**0.67**	**0.33**

Posterior Probability		Play Golf	
		Yes	No
Humidity	High	**0.43**	**0.57**

Posterior Probability		Play Golf	
		Yes	No
Windy			
	True	**0.50**	**0.50**

P(PlayGolf = Yes) = 0.60 x 0.43 x 0.67 x 0.50 = 0.08643

P(PlayGolf = No) = 0.40 x 0.57 x 0.33 x 0.50 = 0.03762

Naïve Bayesian– Multiple Predictors

$$P(T_i \mid \mathrm{X}) = P(x_1 \mid T_i) \times P(x_2 \mid T_i) \times \cdots \times P(x_n \mid T_i) \times P(T_i)$$

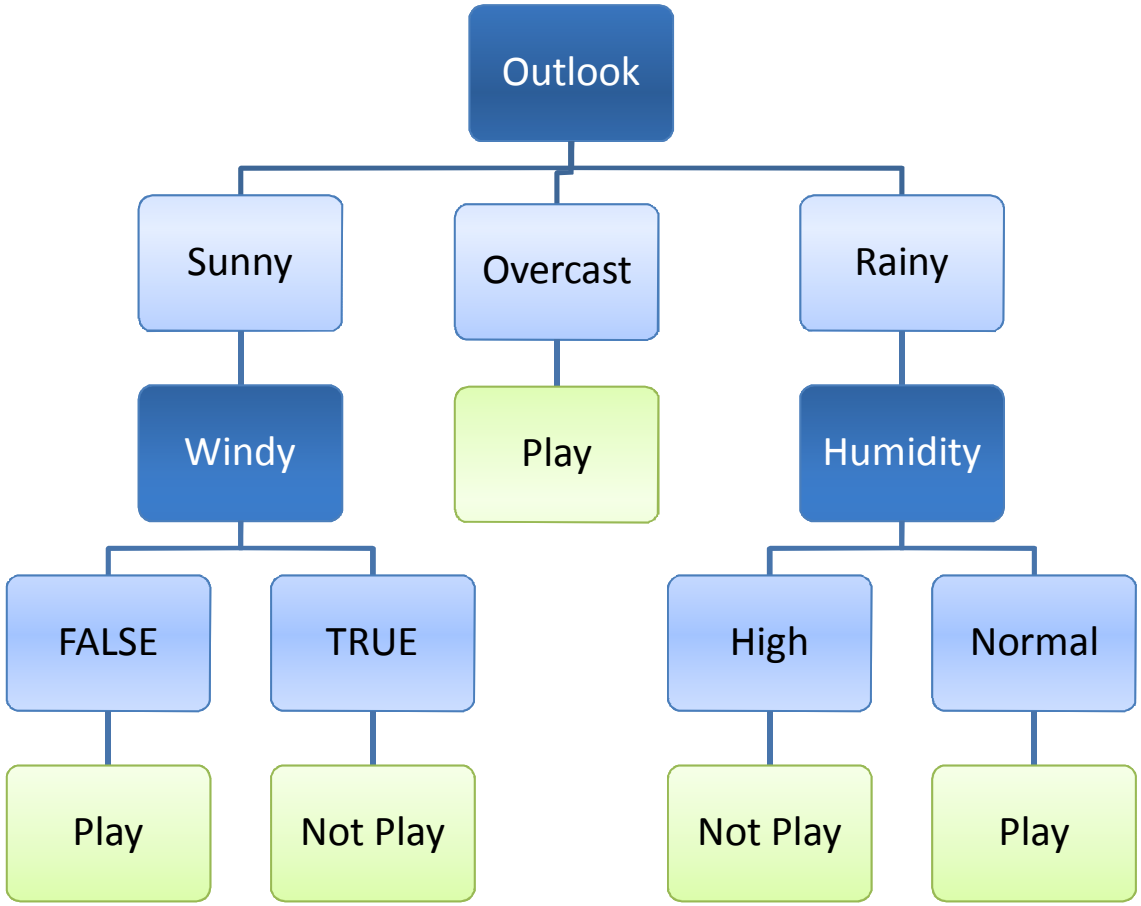
Strong Independence assumption about predictors

Can we incorporate all predictors

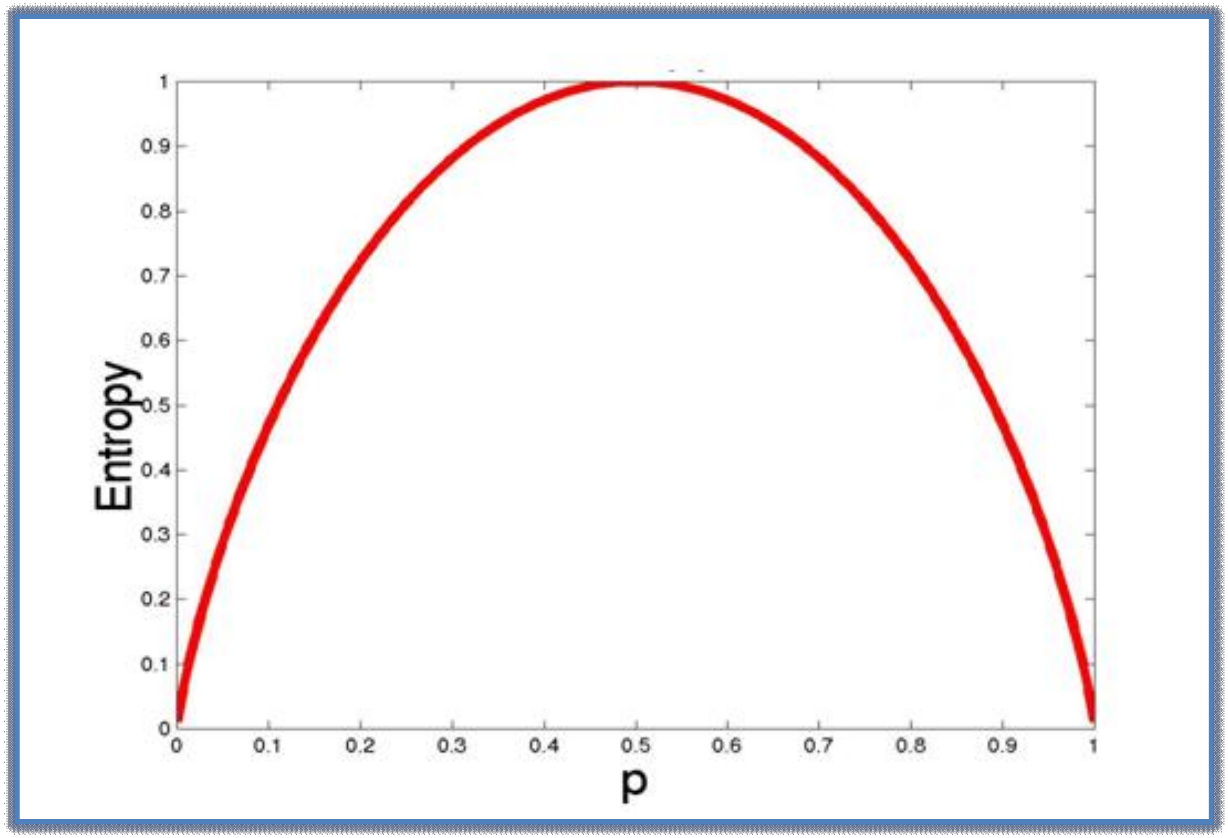
+ Strong dependency?

Decision Trees

Decision Tree



Entropy



Entropy $= -p \log_2 p - q \log_2 q$

Entropy $= -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$

Entropy – Frequency

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Entropy (5,3,2) = Entropy (0.5,0.3,0.2)

= - (0.5 * $\log_2$0.5) - (0.3 * $\log_2$0.3) - (0.2 * $\log_2$0.2)

= 1.49

Entropy - Target

Play Golf
No
No
Yes
Yes
Yes
No
Yes
No
Yes
Yes
Yes
Yes
No

Sort

Play Golf
No
No
No
No
No
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes

5 / 14 = 0.36

9 / 14 = 0.64

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

Frequency Tables

		Play Golf	
		Yes	No
Outlook	Sunny	**3**	**2**
	Overcast	**4**	**0**
	Rainy	**2**	**3**

		Play Golf	
		Yes	No
Temp.	Hot	**2**	**2**
	Mild	**4**	**2**
	Cool	**3**	**1**

		Play Golf	
		Yes	No
Humidity	High	**3**	**4**
	Normal	**6**	**1**

		Play Golf	
		Yes	No
Windy	False	**6**	**2**
	True	**3**	**3**

Entropy – Frequency Table

		Play Golf		
		Yes	No	
Outlook	Sunny	**3**	**2**	**5**
	Overcast	**4**	**0**	**4**
	Rainy	**2**	**3**	**5**
				14

$$E(T,X) = \sum_{v \in X} P(v)E(v)$$

$$E(v) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Information Gain

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

Information Gain – the best predictor?

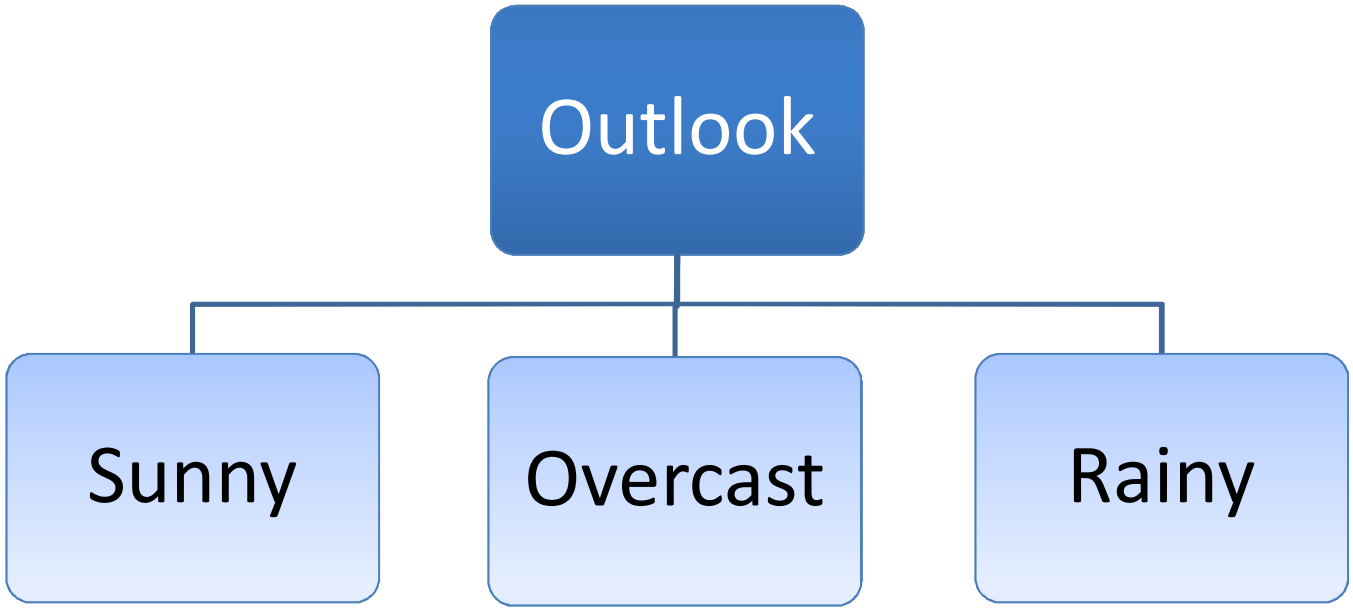
		Play Golf	
		Yes	No
Outlook	Sunny	**3**	**2**
	Overcast	**4**	**0**
	Rainy	**2**	**3**
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	**2**	**2**
	Mild	**4**	**2**
	Cool	**3**	**1**
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	**3**	**4**
	Normal	**6**	**1**
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	**6**	**2**
	True	**3**	**3**
Gain = 0.048			

Decision Tree – Root Node

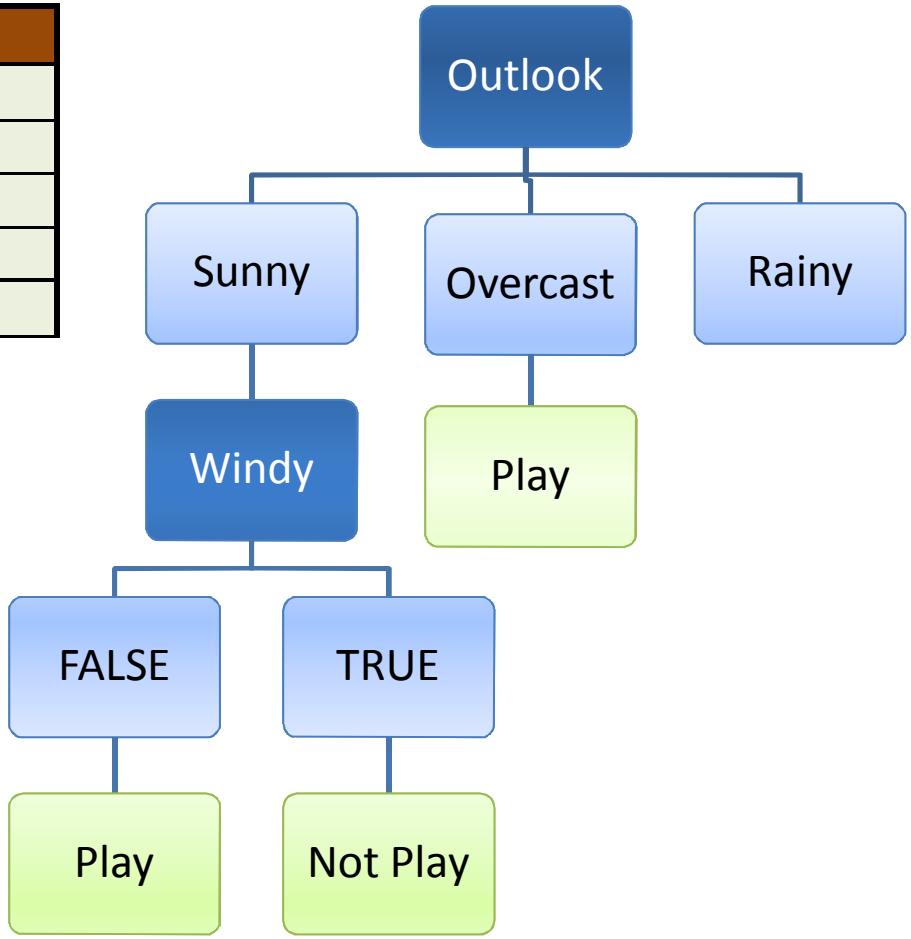


Dataset – Divide and Conqure

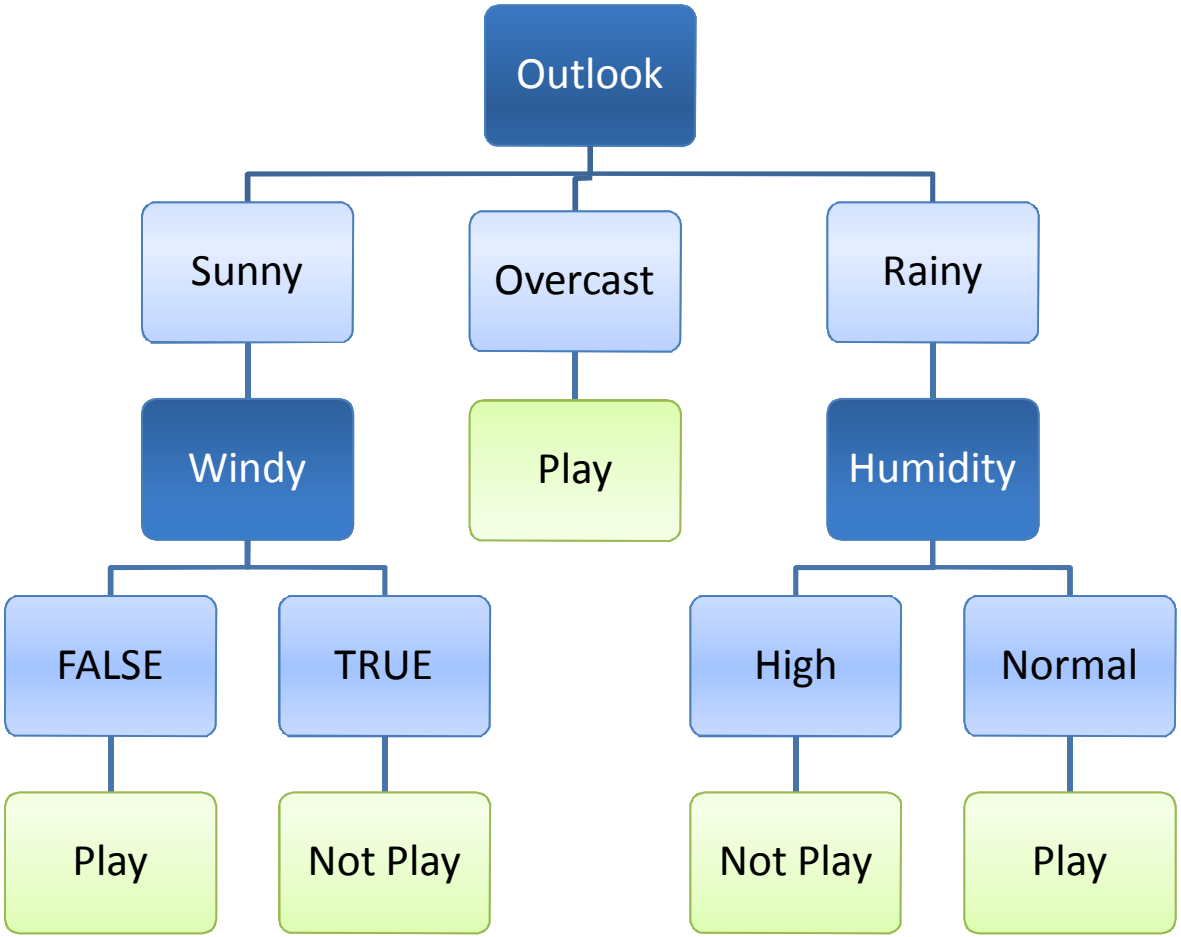
Outlook	Temp.	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes

Subset (Outlook = Sunny)

Temp.	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Final Decision Tree



Numeric Variables and Missing Values

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	85	High	False	No
Rainy	80	High	True	No
Overcast	83	High	False	Yes
Sunny	70	High	False	Yes
Sunny	68	?	False	Yes
Sunny	65	Normal	True	No
Overcast	64	Normal	True	Yes
Rainy	72	High	?	No
Rainy	69	Normal	False	Yes
Sunny	75	Normal	False	Yes
Rainy	75	Normal	True	Yes
?	72	High	True	Yes
Overcast	81	Normal	False	Yes
Sunny	71	High	True	No

Numeric Variables - Binning

Temp	B_Temp	Play Golf
85	80-90	No
80	80-90	No
83	80-90	Yes
70	70-80	Yes
68	60-70	Yes
65	60-70	No
64	60-70	Yes
72	70-80	No
69	60-70	Yes
75	70-80	Yes
75	70-80	Yes
72	70-80	Yes
81	80-90	Yes
71	70-80	No

		Play Golf	
		Yes	No
B_Temp	60-70	3	1
	70-80	4	2
	80-90	2	2

Models based on Frequency Table

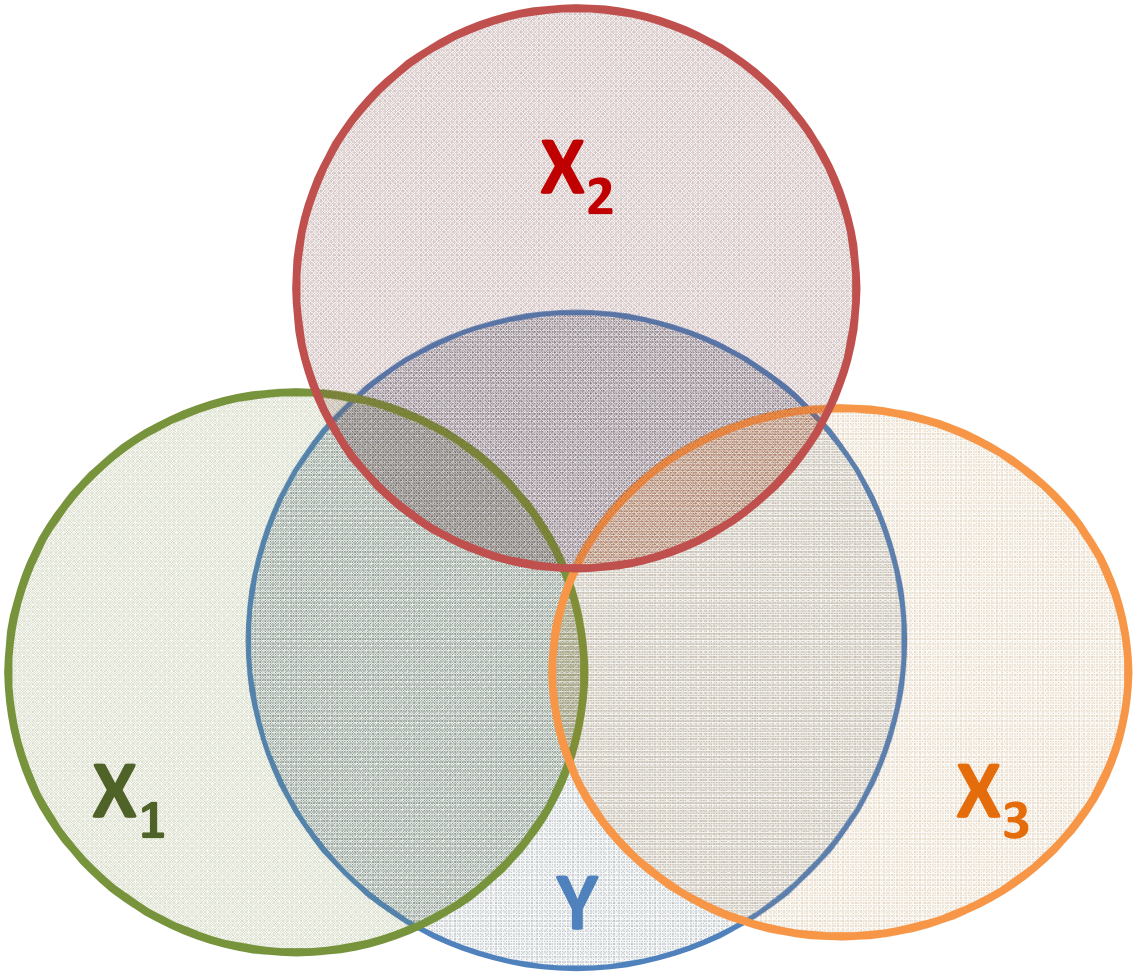
	Robust	Measurable	Scalable
OneR	✓	✓	✓
Naïve Bayesian	✓	✓	✓
Decision Tree	✓	✓	✗

Models

based on

Covariance Matrix

Models based on Covariance Matrix



Covariance

Covariance matrix

$$C = \frac{\sum (x - \mu_x)(y - \mu_y)}{n}$$

Mean

$$\mu = \frac{\sum x}{n}$$

Multiple Linear Regression
MLR

$$Y = \beta_o + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$$

Find β by minimizing ε

MLR : Ordinary Least Square

Intercept and Slopes:

$$\beta = (X'X)^{-1} X'Y$$

$$b = \frac{Cov\,(x, y)}{Var\,(x)}$$

Predicted Values:

$$Y' = \beta X$$

Residuals:

$$Y - Y'$$

Regression Statistics

$$SST = \sum (Y - \overline{Y})^2$$

$$SSR = \sum (Y' - \overline{Y'})^2$$

$$SSE = \sum (Y - Y')^2$$

Regression Statistics

How good is our model?

$$R^2 = \frac{SSR}{SST}$$

Coefficient of Determination

to judge the adequacy of the regression model

ANOVA

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_k = 0$$

$$H_A : \beta_i \neq 0$$ *at least one!*

	df	SS	MS	F	P-value
Regression	*k*	*SSR*	*SSR / df*	*MSR / MSE*	*P(F)*
Residual	*n-k-1*	*SSE*	*SSE / df*		
Total	*n-1*	*SST*			
If P(F)<a then we know that we get significantly better prediction of Y from the regression model than by just predicting mean of Y.					

Hypotheses Tests for Regression Coefficients

$$H_0 : \beta_i = 0$$
$$H_A : \beta_i \neq 0$$

$$t_{(n-k-1)} = \frac{b_1 - \beta_i}{S_e(b_i)} = \frac{b_i - \beta_i}{\sqrt{S_e^2 C_{ii}}}$$

$$\mathbf{C} = (\mathbf{X}'\mathbf{X})^{-1} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix}$$

Multicolinearity

- If the F-test for significance of regression is significant, but tests on the individual regression coefficients are not, multicolinearity may be present.

- Variance Inflation Factors (VIFs) are very useful measures of multicolinearity. If any VIF exceed 5, multicolinearity is a problem.

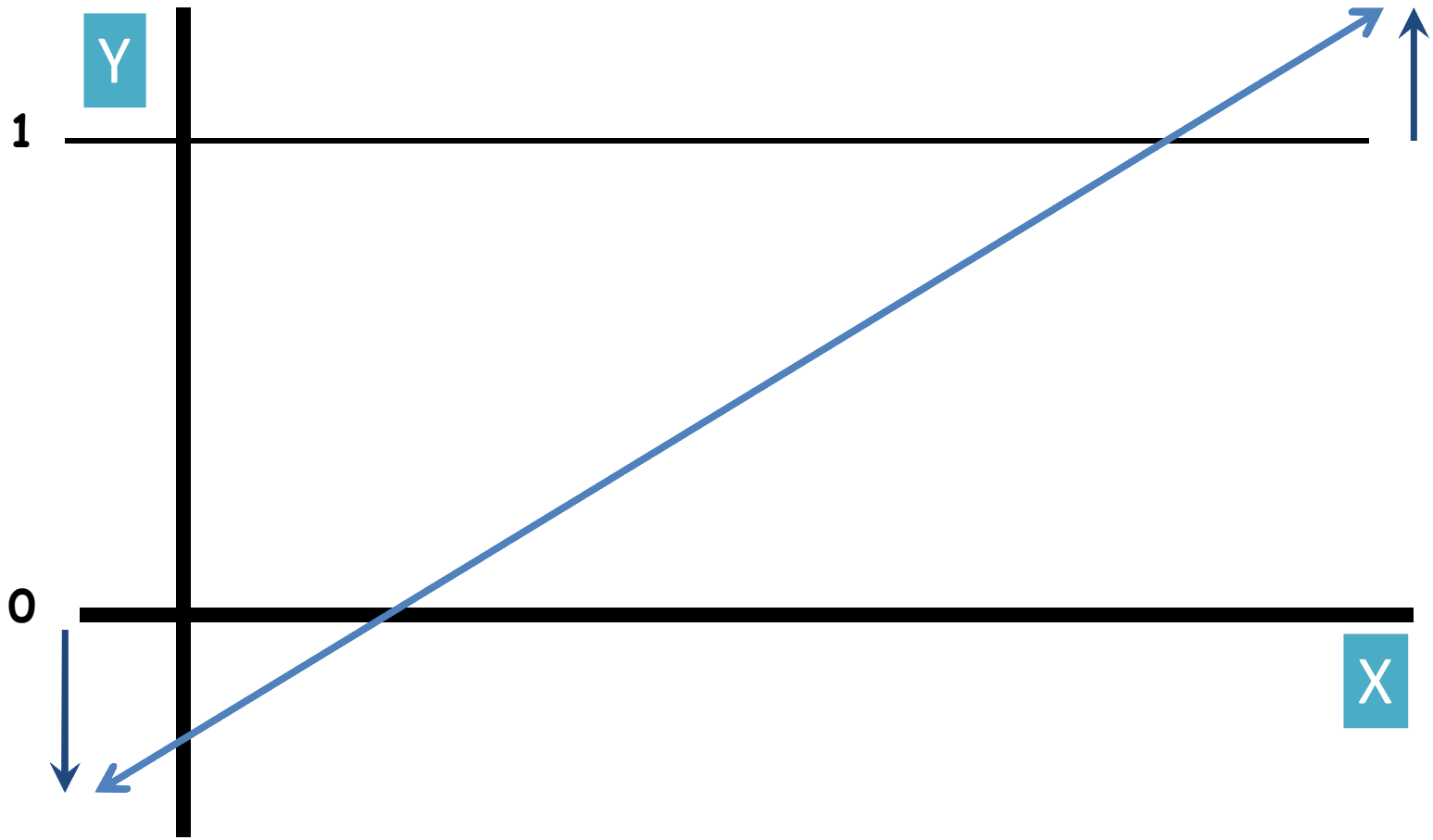
$$VIF(\beta_i) = \frac{1}{1 - R_i^2} = C_{ii}$$

Linear Regression – Binary Target

$$Y_i = \beta_o + \beta_1 X_i + \varepsilon_i$$

- If actual Y is a binary variable the predicted Y can be less than zero or greater than 1.
- If actual Y is a binary variable ***error is not normally distributed.***

Linear Regression Model



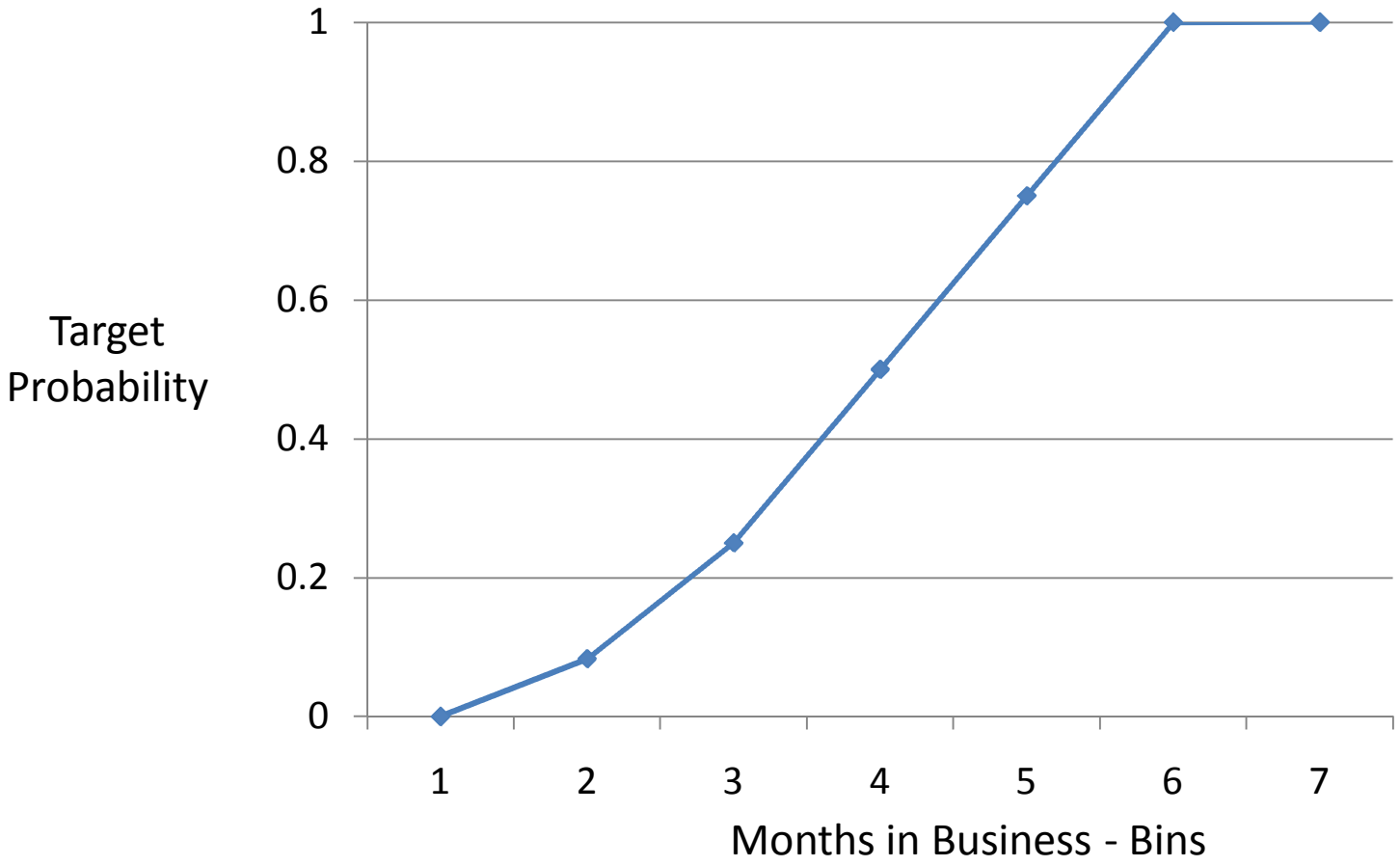
Logistic Regression

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

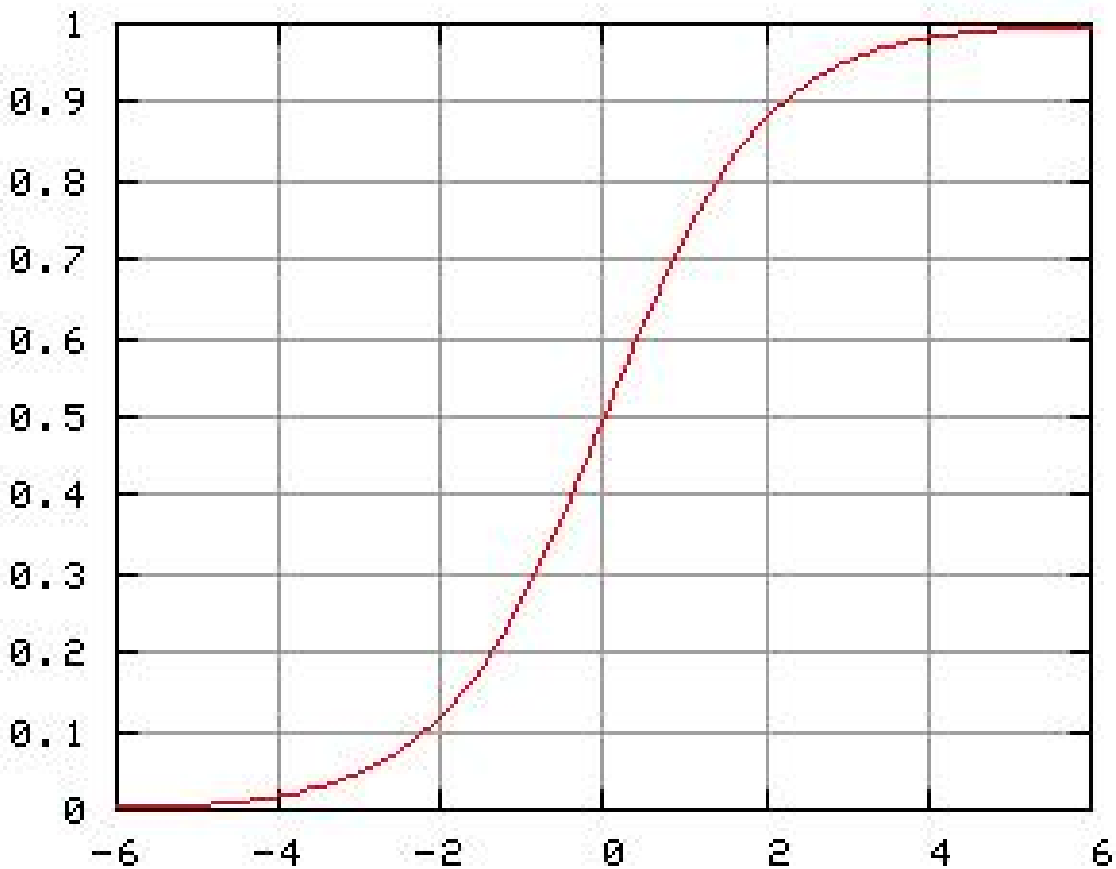
Frequency Table

Months in Business	Count	Target Count	Target Probability
<50	4	0	0
50-100	12	1	0.083
100-150	4	1	0.25
150-200	4	2	0.5
200-250	4	3	0.75
250-300	1	1	1
>300	4	4	1

Frequency Plot



Logistic Function



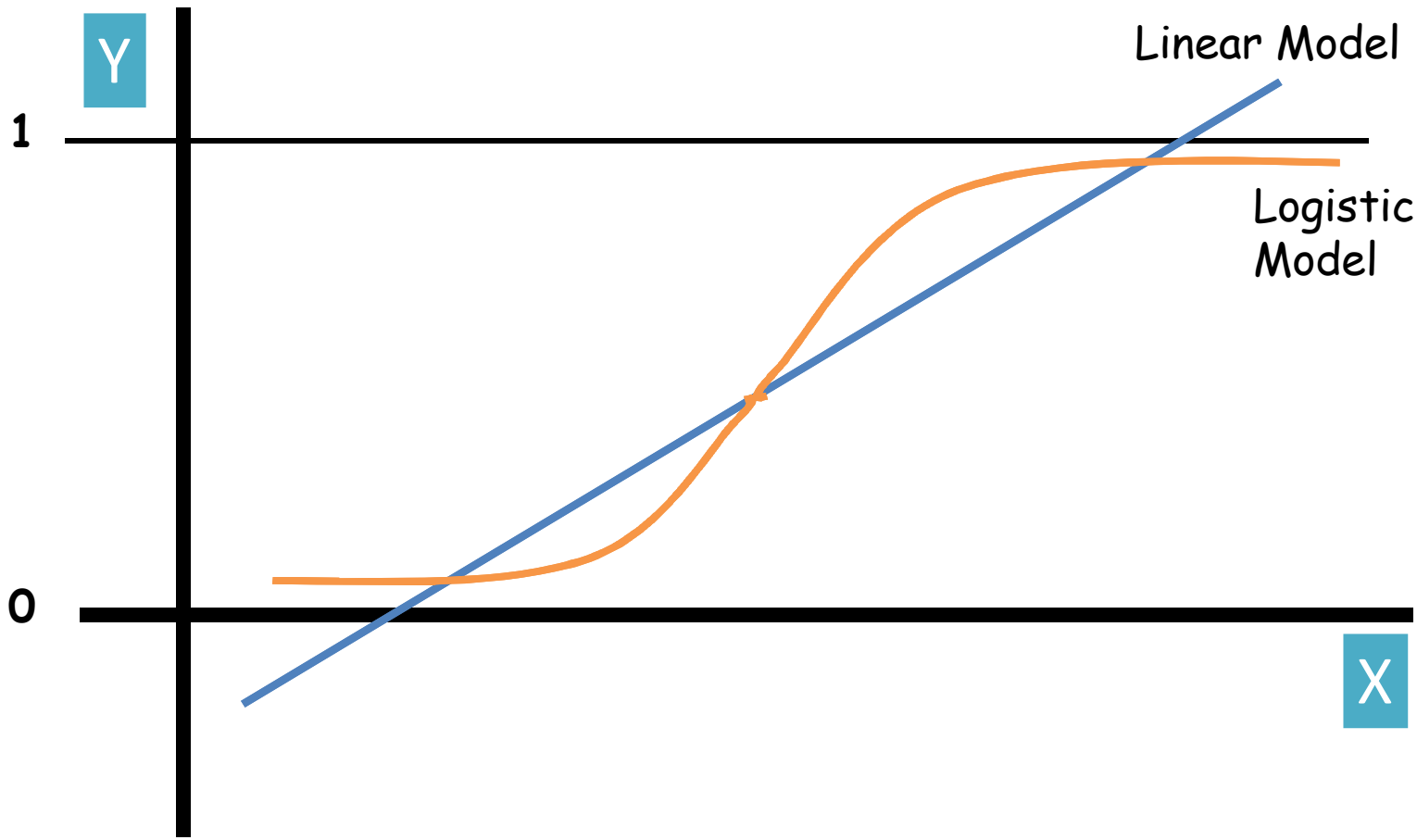
$$f(z) = \frac{1}{1+e^{-z}}$$

Logistic Regression

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

- The logistic distribution constrains the estimated probabilities to lie between 0 and 1.
- *Maximum Likelihood Estimation* is a statistical method for estimating the coefficients of a model.

Logistic Regression Model



Log Likelihood (LL)

- Likelihood is the probability that the dependent variable may be predicted from the independent variables.
- LL is calculated through *iteration*, using maximum likelihood estimation (MLE).
- Log likelihood is the basis for tests of a logistic model.

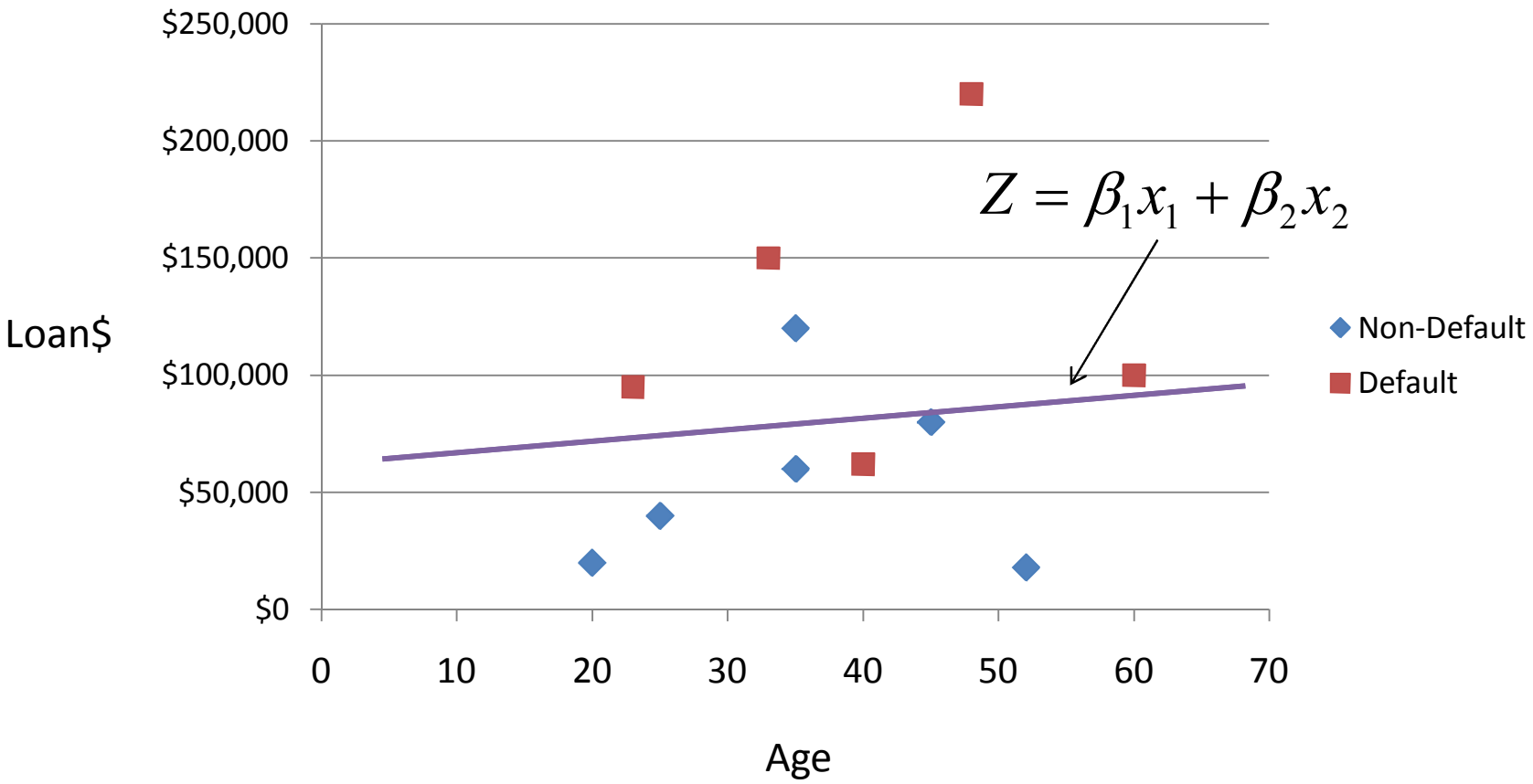
Wald Test

- A Wald test is used to test the statistical significance of each coefficient (β) in the model.
- A Wald test calculates a *Z* statistic, which is:

$$Z = \frac{\hat{\beta}}{SE}$$

- This *Z* value is then squared, yielding a Wald statistic with a chi-square distribution.

Linear Discriminant Analysis - LDA



LDA – Training

$$Z = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

$$S(\beta) = \frac{\beta^T \mu_1 - \beta^T \mu_2}{\beta^T C \beta}$$

Score function

$$S(\beta) = \frac{\overline{Z}_1 - \overline{Z}_2}{\text{Variance of Z within groups}}$$

LDA – Training

Mean

$$\mu = \frac{\sum x}{n}$$

Covariance matrix

$$C = \frac{\sum (x - \mu_x)(y - \mu_y)}{n}$$

LDA – Training

$$\beta = C^{-1}(\mu_1 - \mu_2)$$

$$C = \frac{1}{n_1 + n_2}(n_1 C_1 + n_2 C_2)$$

β : *Linear model coefficients*

C : *Pooled covariance matrix*

μ_1, μ_2 : *Mean vectors*

LDA – Model Assessment

$$\Delta^2 = \beta^T (\mu_1 - \mu_2)$$

Δ : *Mahalanobis distance between two groups*

LDA – Prediction

$$Z_0 = \beta^T \left(\frac{\mu_1 + \mu_2}{2} \right)$$

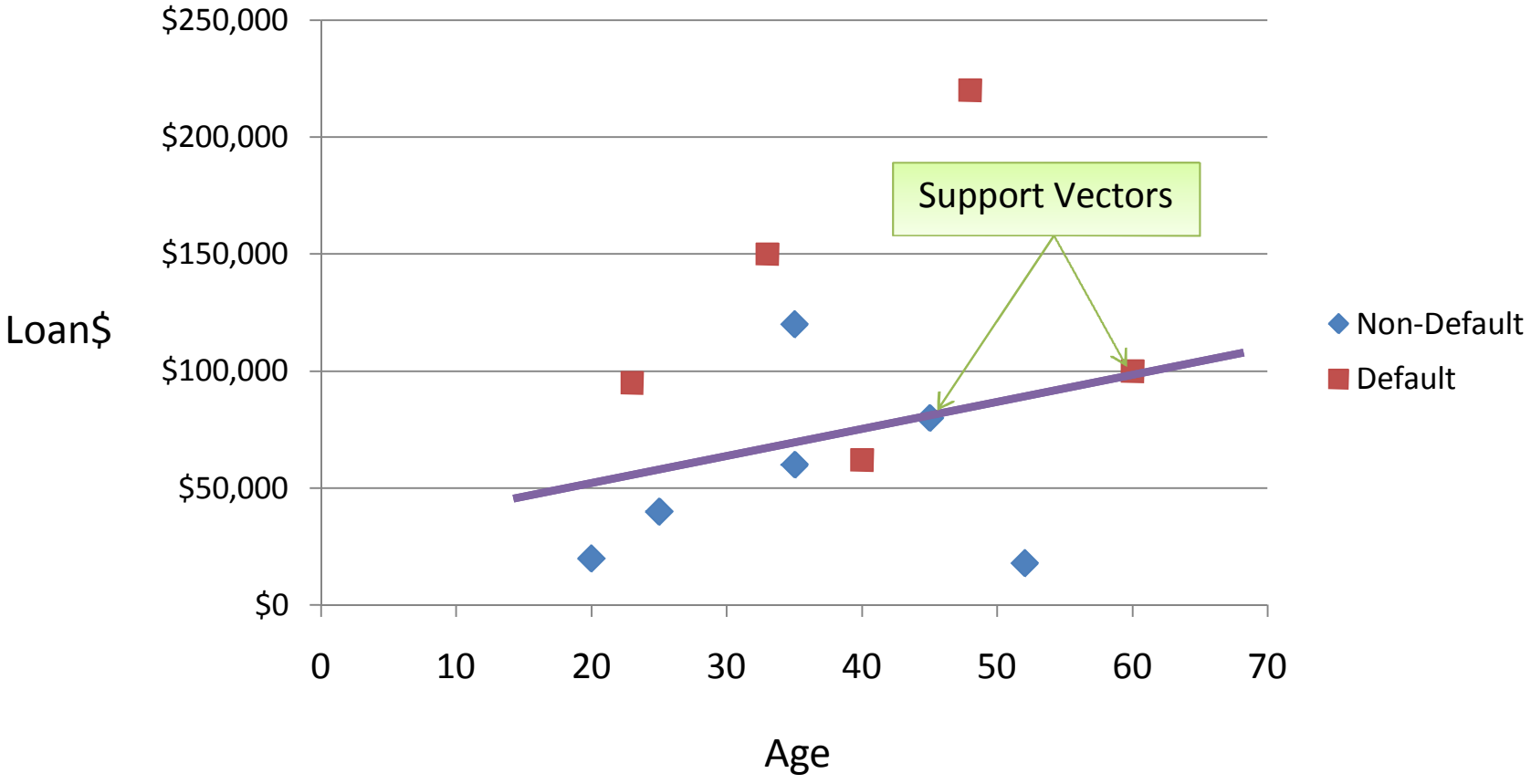
$$\overline{Z}_1 = \beta^T \mu_1$$

$$\overline{Z}_2 = \beta^T \mu_2$$

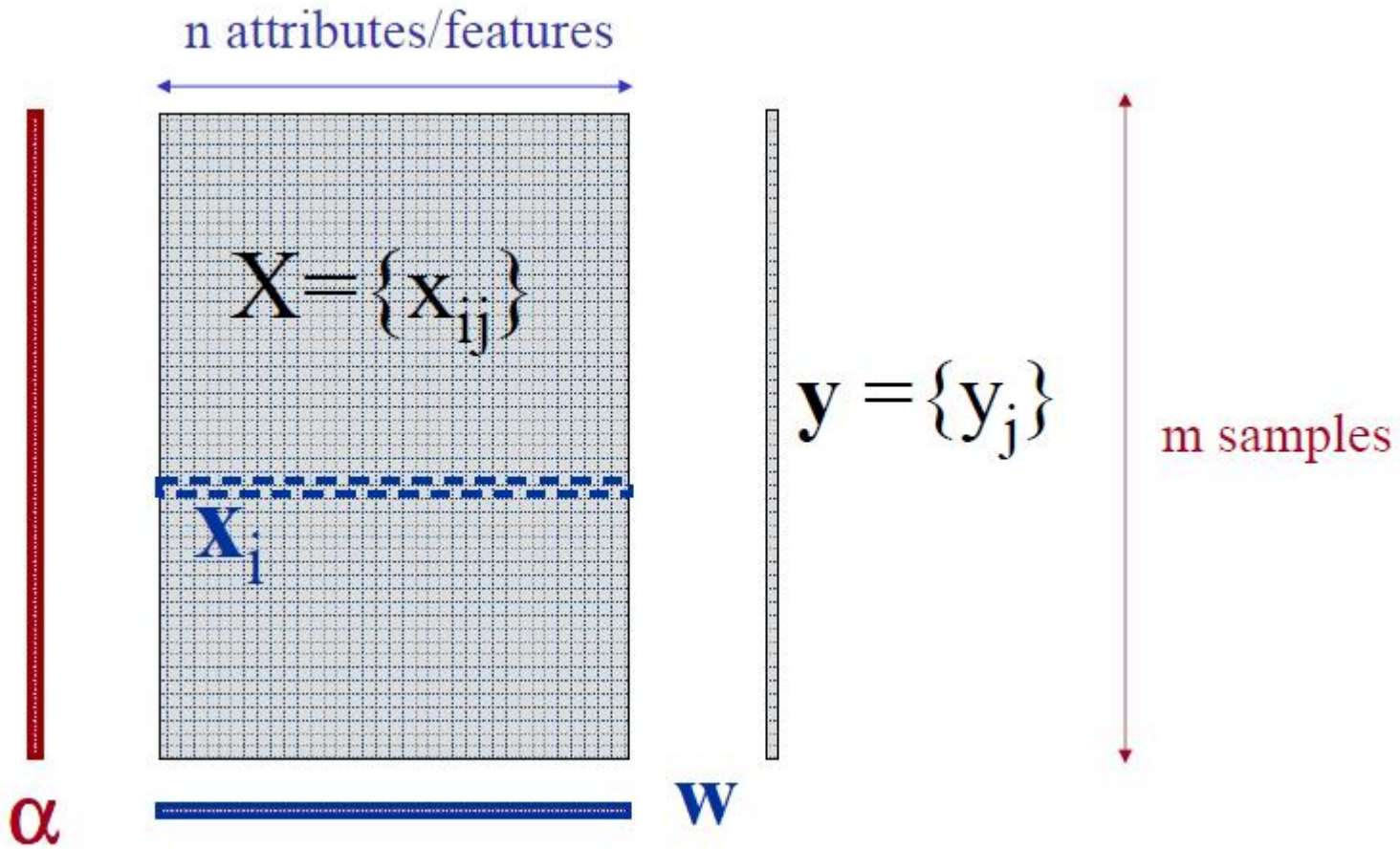
Can we handle non-linearity?

Support Vector Machines

Support Vector Machines- SVM



SVM – Dual Form



SVM – Dual Form

$$f(x) = \sum_{j=1}^{n} w_j x_j + b$$

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i x) + b$$

SVM – Kernel functions

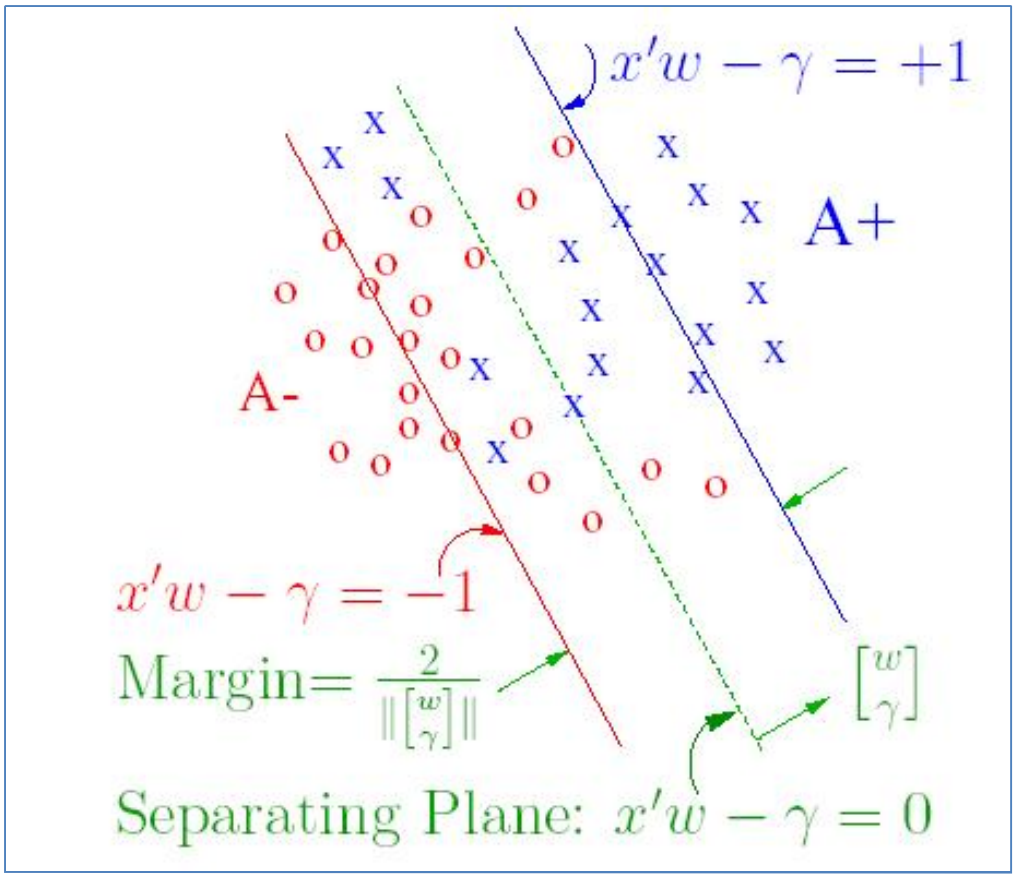
Polynomial

$$k(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}_i . \mathrm{x}_j)^d$$

Gaussian Radial Basis function

$$k(\mathrm{x}_i, \mathrm{x}_j) = \exp\left(-\frac{\left\| \mathrm{x}_i - \mathrm{x}_j \right\|^2}{2\sigma^2} \right)$$

Linear SVM



Linear Proximal SVM Algorithm defined by Fung and Mangasarian: *Given m data points in R^n represented by the **m x n matrix A** and a diagonal **D of +1 and -1** labels denoting the class of each row of A, we generate the linear classifier as follows:*

$$sign(x'\omega - \gamma)\begin{cases} = 1, then\ x \in A+ \\ = -1, then\ x \in A- \end{cases}$$

$$E = \begin{bmatrix} A & -e \end{bmatrix}$$

where e is an m x 1 vectors of ones.

$$\begin{bmatrix} \omega \\ \gamma \end{bmatrix} = (\frac{I}{\nu} + E'E)^{-1} E'De$$

Typically v is chosen by means of a tuning (validating) set.

Models based on Covariance Matrix

	Robust	Measurable	Scalable
MLR	✓	✓	✓
LDA	✓	✓	✓
Logistic Reg.	✓	✓	✗
SVM	✓	✓	✗

Models

based on

Similarity

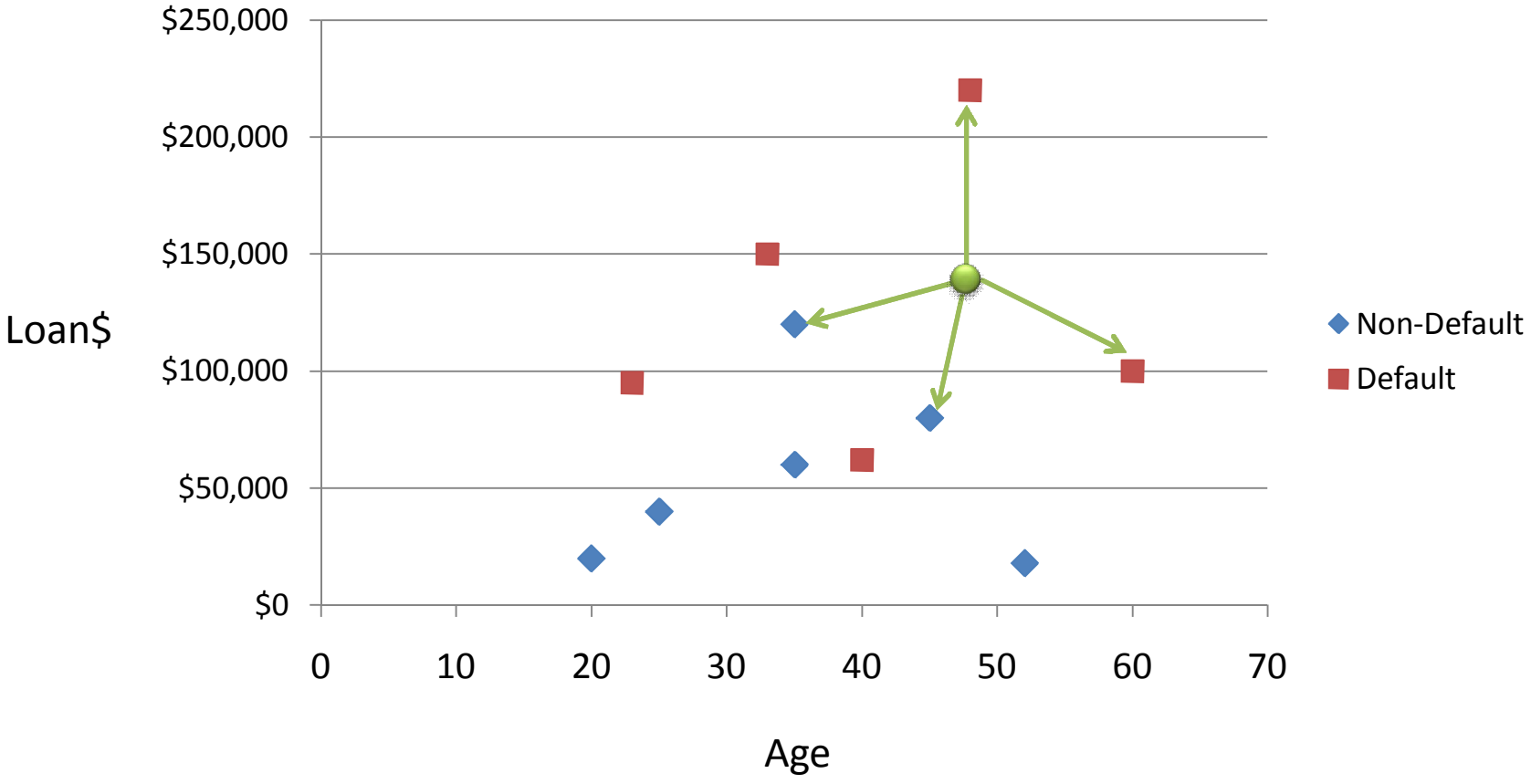
KNN - Definition

KNN is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure.

KNN – different names

- K-Nearest Neighbors
- Memory-Based Reasoning
- Example-Based Reasoning
- Instance-Based Learning
- Case-Based Reasoning
- Lazy Learning

KNN Classification



KNN Classification – Distance

Age	Loan	Default	Distance
25	$40,000	N	102000
35	$60,000	N	82000
45	$80,000	N	62000
20	$20,000	N	122000
35	$120,000	N	22000
52	$18,000	N	124000
23	$95,000	Y	47000
40	$62,000	Y	80000
60	$100,000	Y	42000
48	$220,000	Y	78000
33	$150,000	Y	**8000**
48	**$142,000**	**?**	

Euclidean Distance

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

KNN Classification – Standardized Distance

Age	Loan	Default	**Distance**
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	**0.3160**
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
0.7	**0.61**	**?**	

$$X_s = \frac{X - Min}{Max - Min}$$

KNN – Number of Neighbors

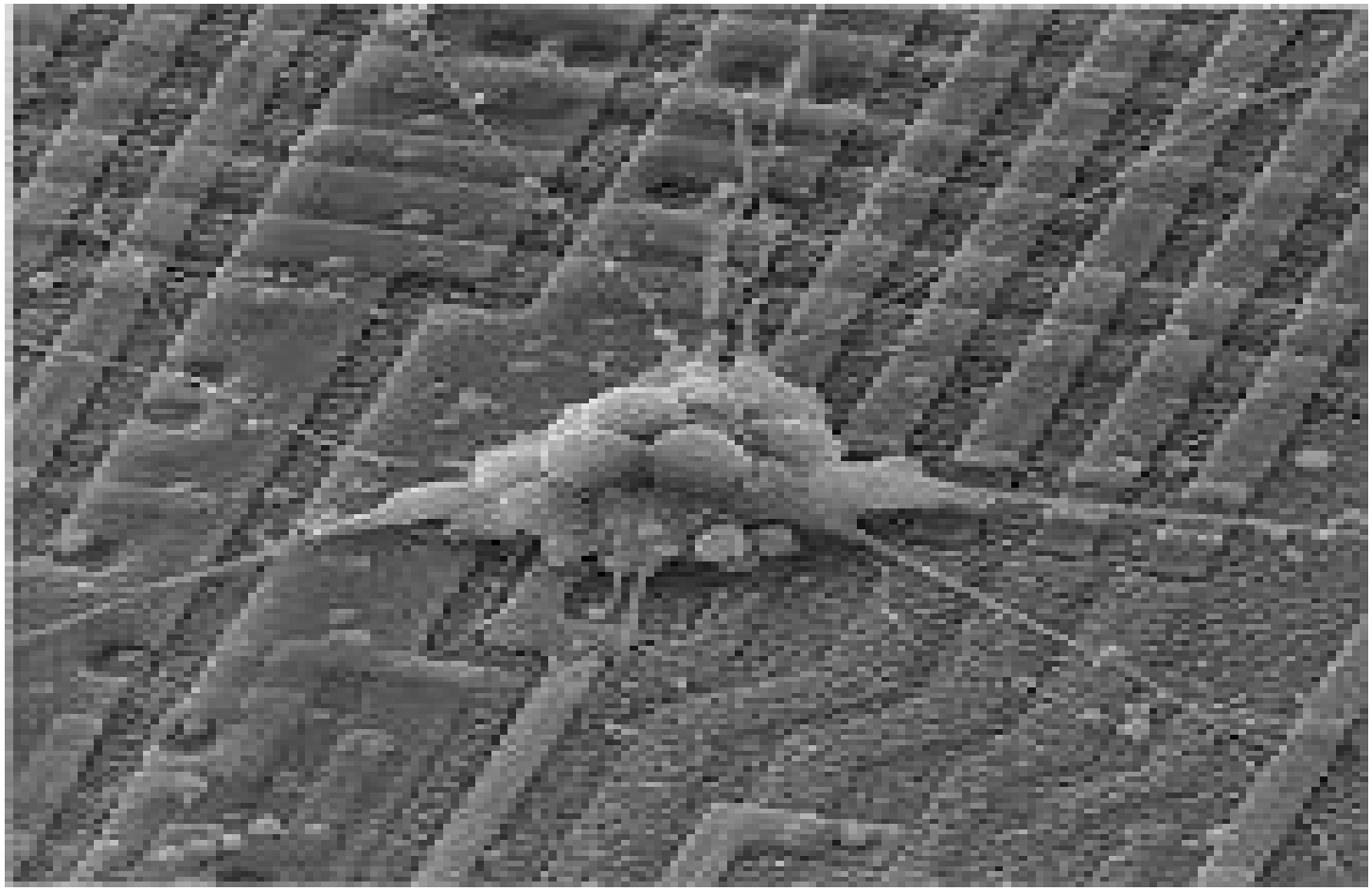
- If K=1, select the nearest neighbor
- If K>1,
  - For classification select the most frequent neighbor.
  - For regression calculate the average of K neighbors.
  - What is the optimal K?

Models based on Similarity

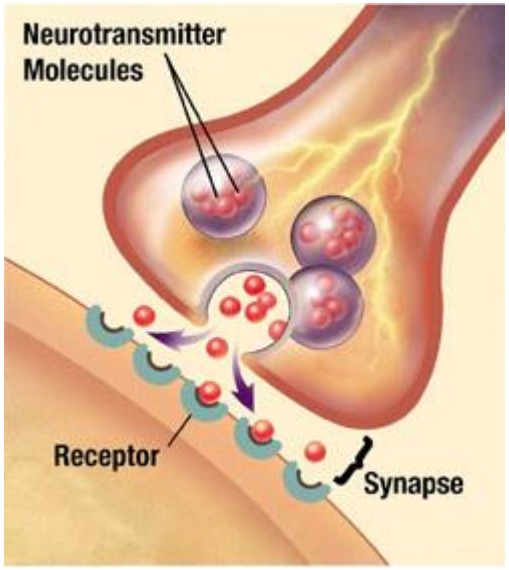
	Robust	Measurable	Scalable
KNN	✓	✗✓	✗✓

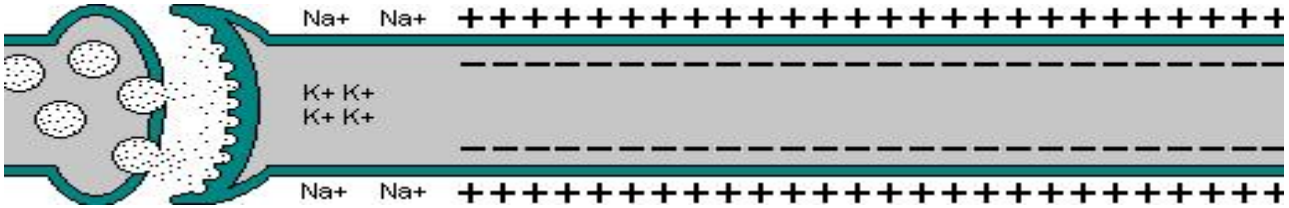
Neural Networks

Biological Neuron and Integrated Circuit

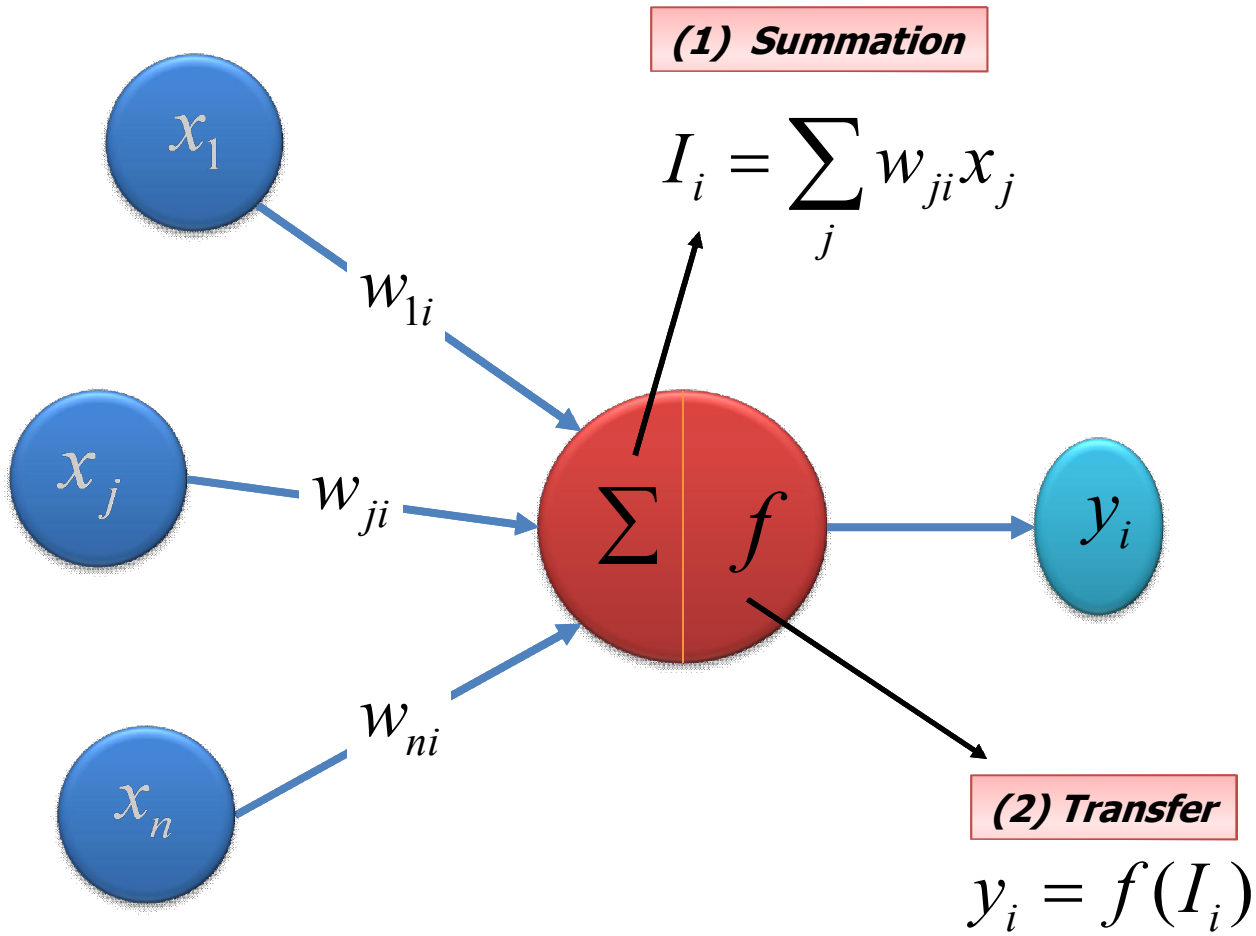


Biological Neuron Synapse

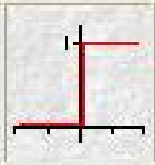
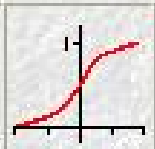
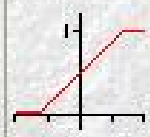
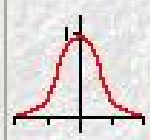




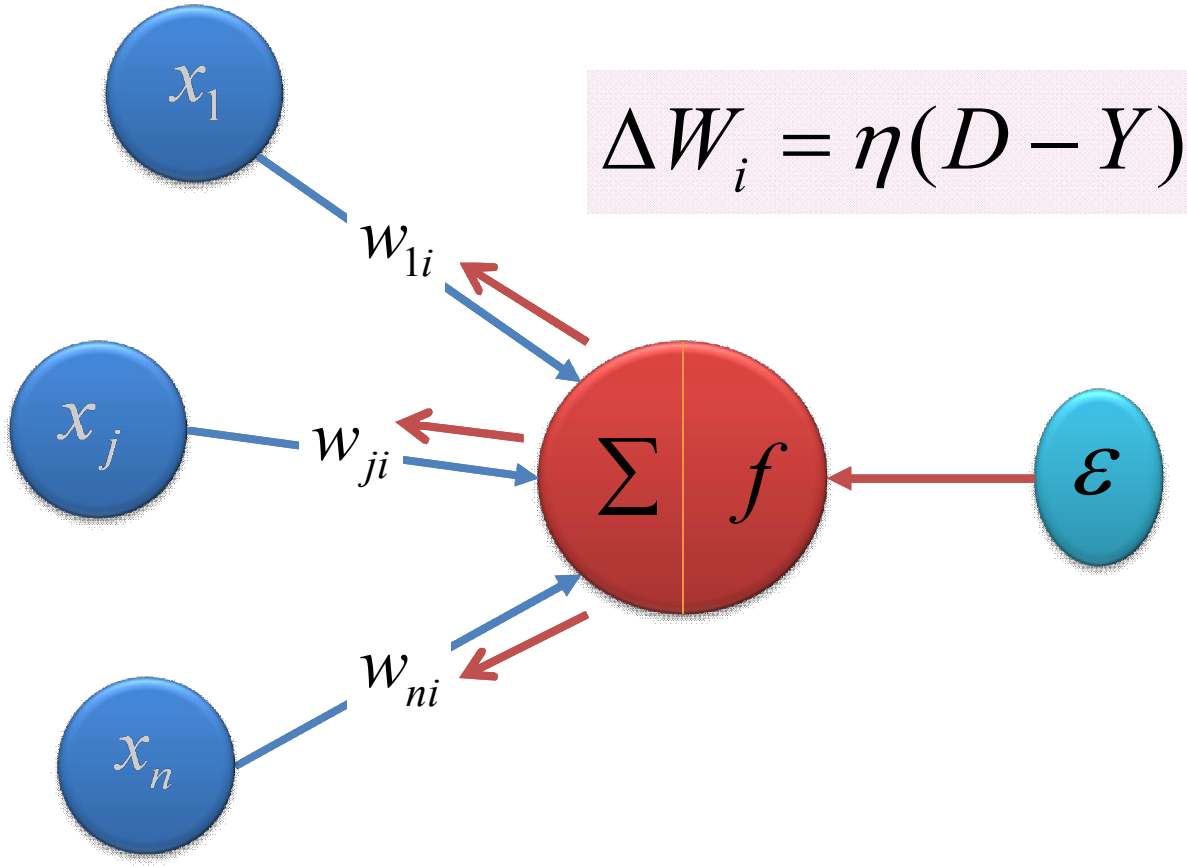
Neural Network - Neuron



Transfer Functions

Unit Step		$f(x) = \begin{cases} 0 \text{ if } 0 > x \\ 1 \text{ if } x \geq 0 \end{cases}$
Sigmoid		$f(x) = \frac{1}{1+e^{-\beta x}}$
Piecewise Linear		$f(x) = \begin{cases} 0 & \text{if } x \leq x_{min} \\ mx+b & \text{if } x_{max} > x > x_{min} \\ 1 & \text{if } x \geq x_{max} \end{cases}$
Gaussian		$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$

Weight Adjustment or Error Propagation

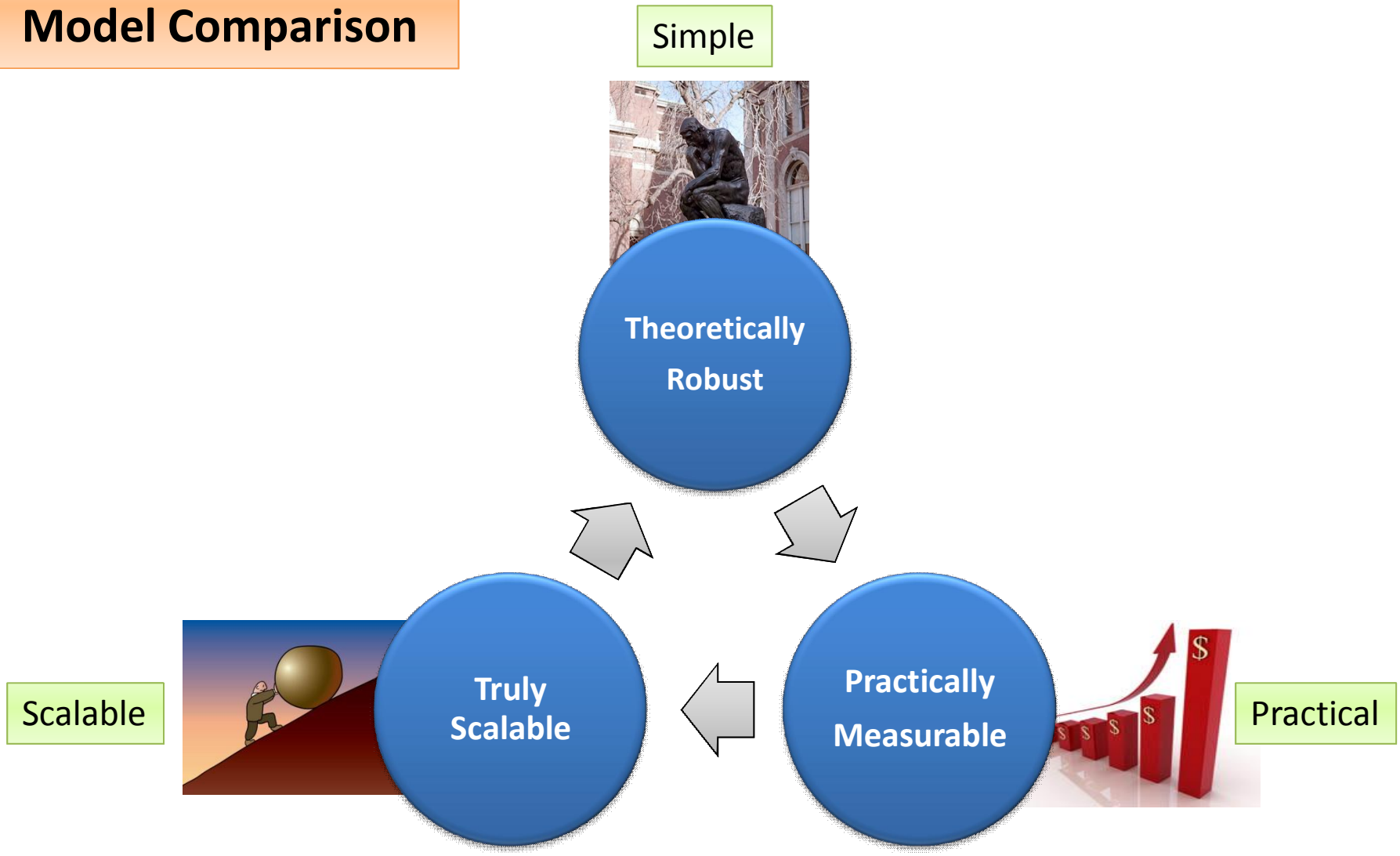


$$\Delta W_i = \eta(D - Y)X_i$$

Neural Networks

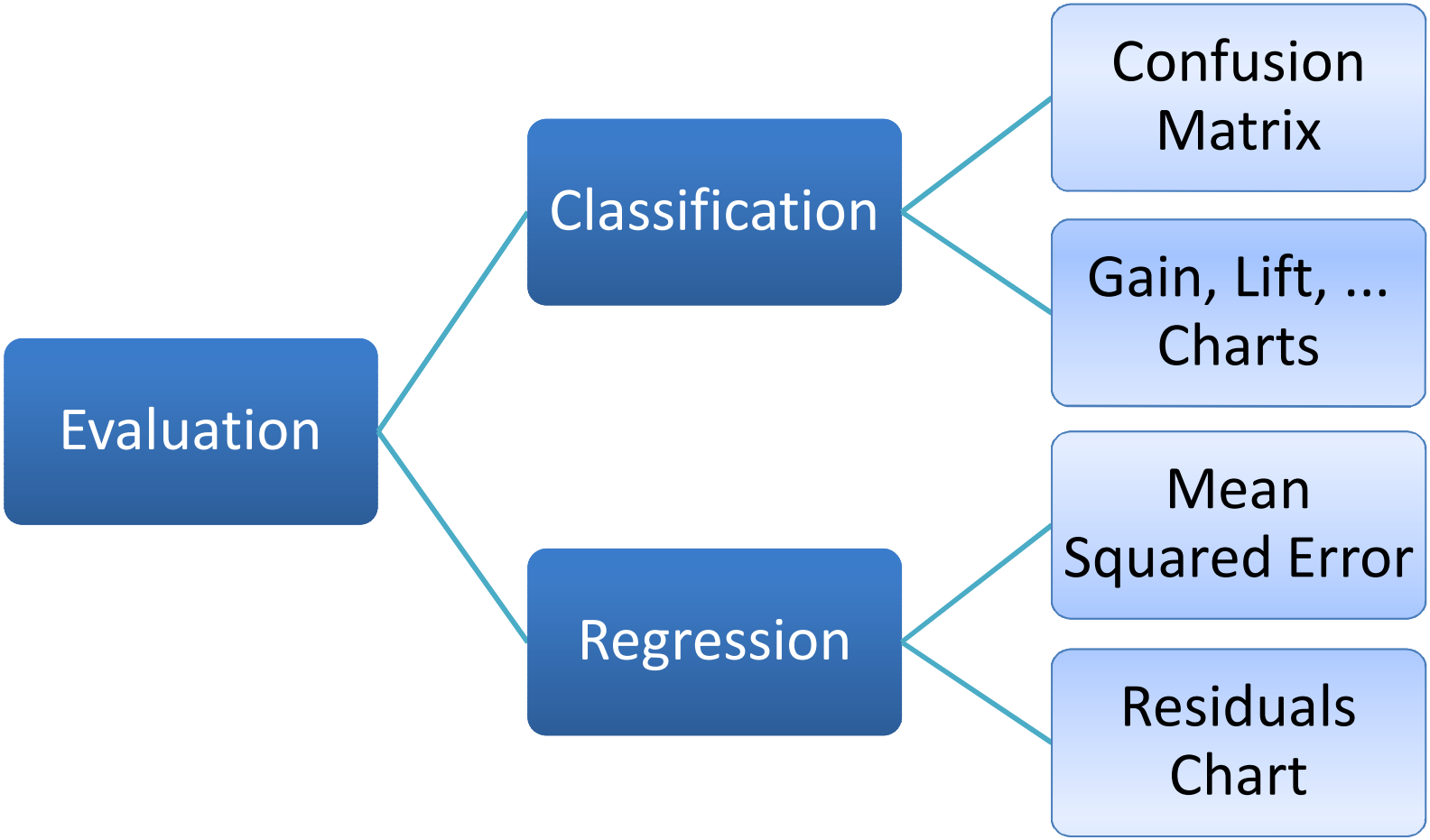
	Robust	Measurable	Scalable
Neural Networks	✓	✗✓	✗

Model Comparison

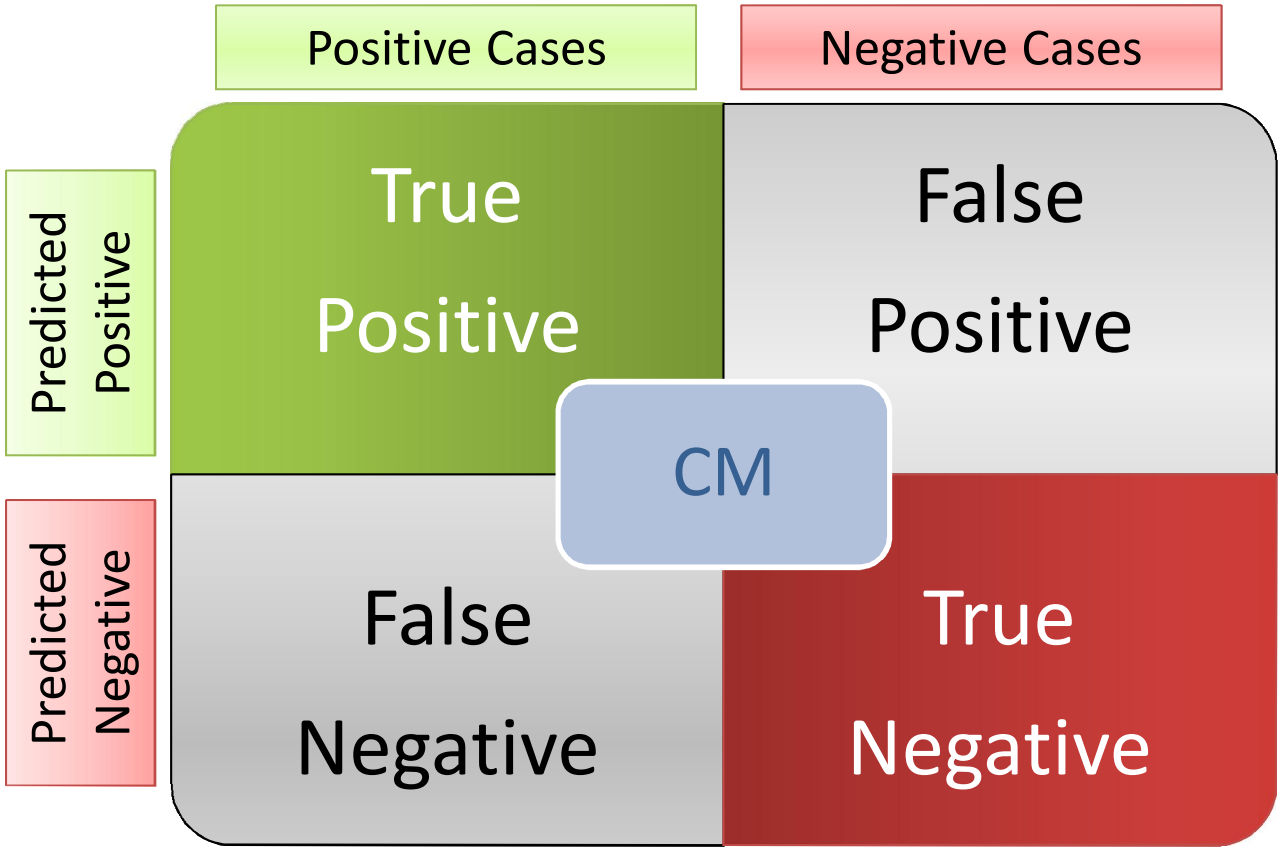


Measurable
(Model Evaluation)

Model Evaluation



Classification - Confusion Matrix



Classification Matrix

		Target		
		Y	N	
Model	Y	***TP***	***FP***	**Positive Predictive Value** TP/(TP+FP)
	N	***FN***	***TN***	**Negative Predictive Value** TN/(TN+FN)
		Sensitivity TP/(TP+FN)	**Specificity** TN/(TN+FP)	**Accuracy** $\frac{TP + TN}{TP + TN + FP + FN}$

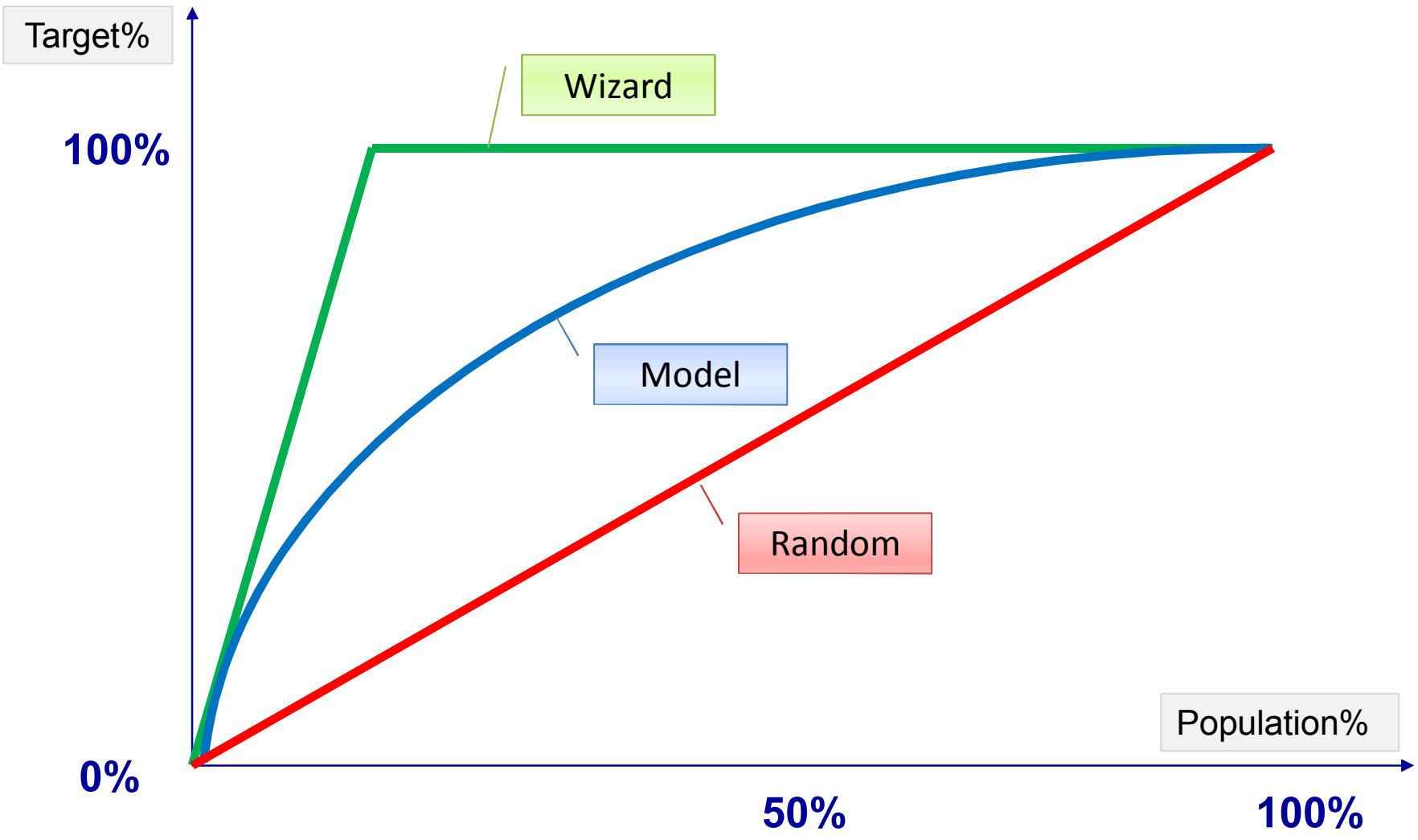
Classification Matrix - OneR

OneR		Target		
		Y	N	
Model	Y	129	431	**Positive Predictive Value** 0.23
	N	79	7528	**Negative Predictive Value** 0.99
		Sensitivity 0.62	**Specificity** 0.95	**Accuracy** 0.94

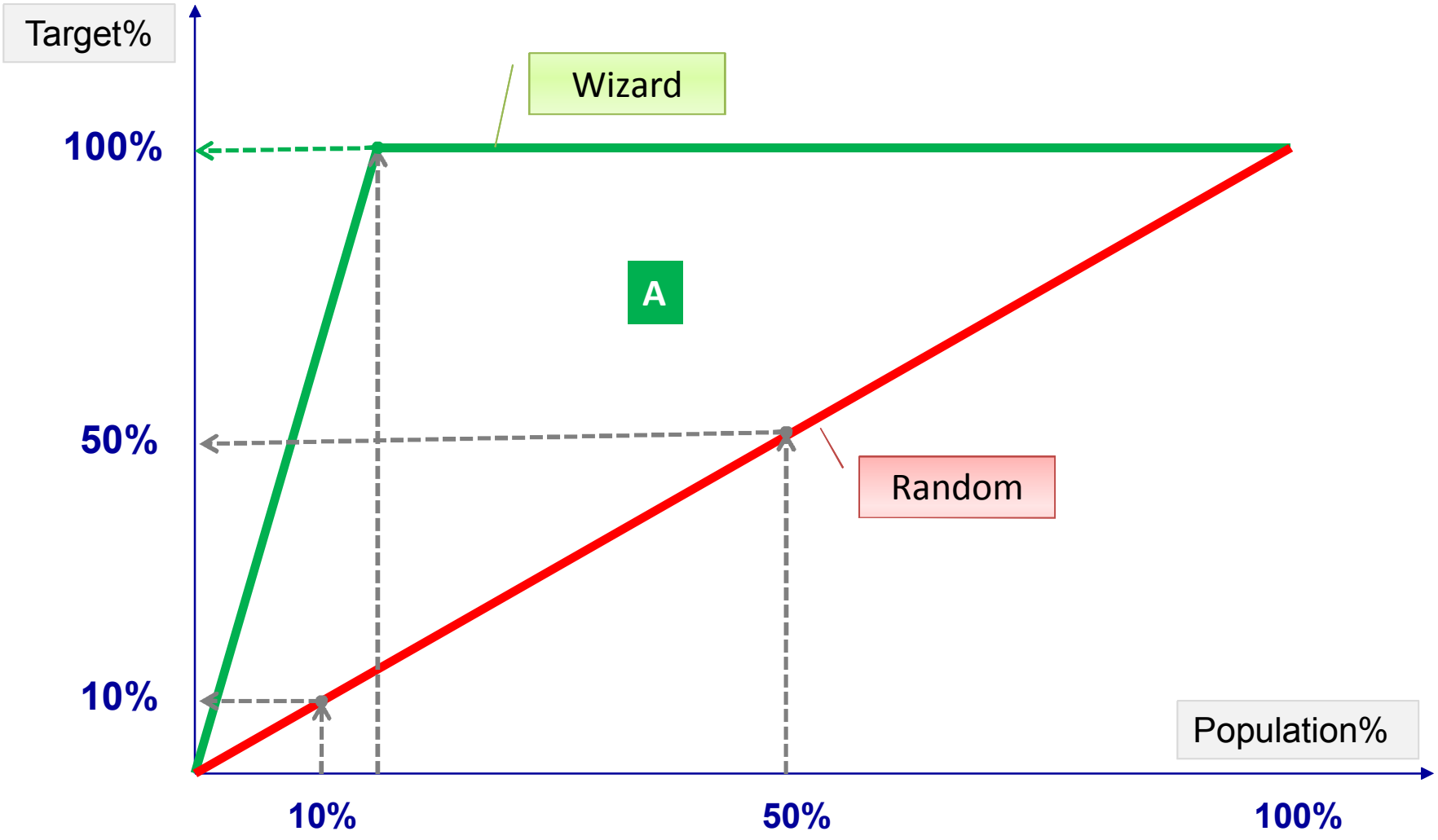
Classification Matrix – Logistic Regression

LogReg		Target		
		Y	N	
Model	Y	135	425	**Positive Predictive Value** 0.24
	N	50	7557	**Negative Predictive Value** 0.99
		Sensitivity 0.73	**Specificity** 0.95	**Accuracy** 0.94

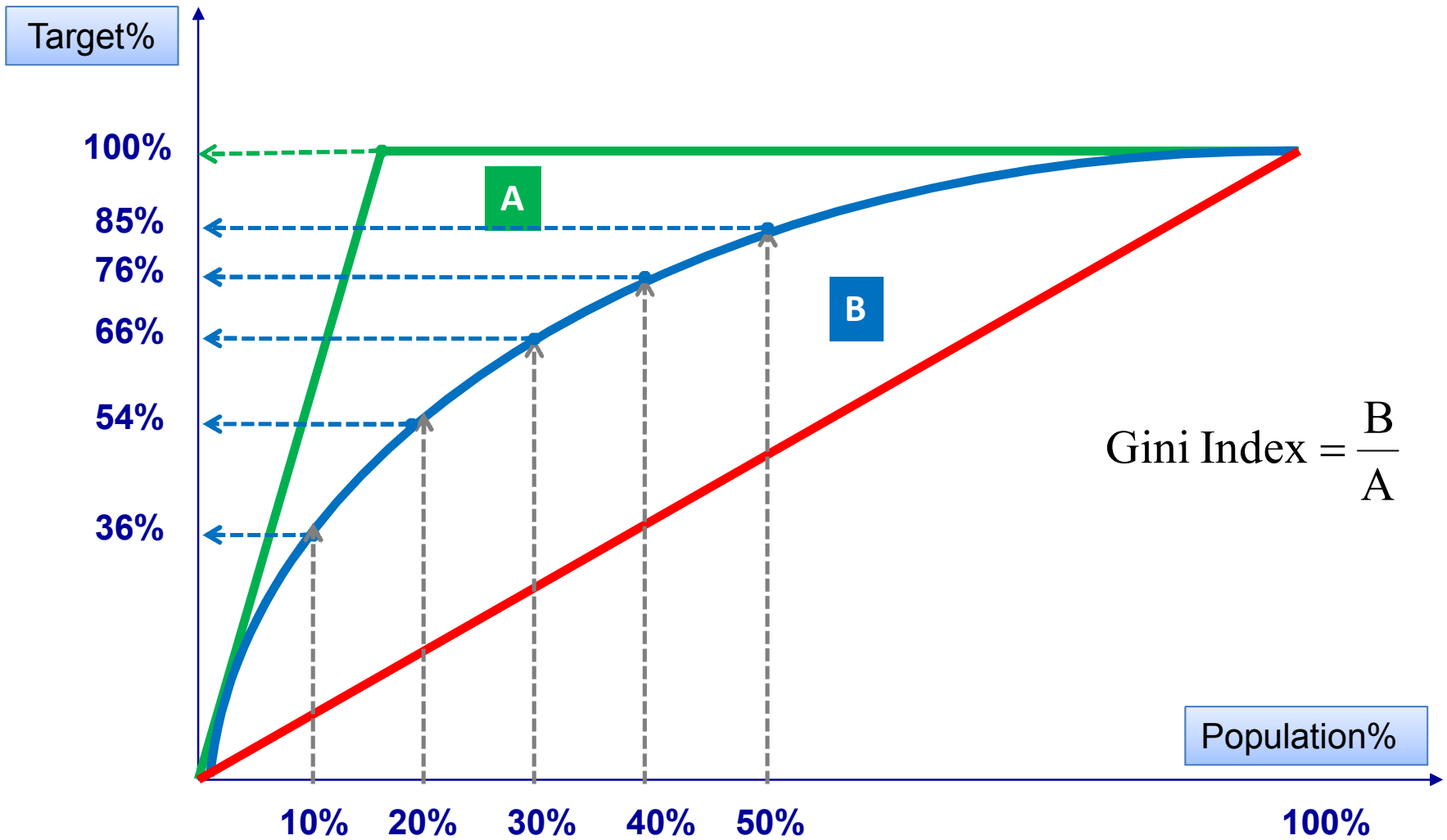
Classification – Gain Chart



Gain Chart



Classification – Gain Chart



$$\text{Gini Index} = \frac{B}{A}$$

Gain Chart

Sorted by Score

Target	Score
1	880
1	724
1	676
1	556
0	480
0	368
0	345
0	235
0	195
...	...

Gain Table

Population%	Target%
10	36
20	54
30	66
40	76
50	85
60	90
70	94
80	98
90	100
100	100

Lift Chart

Gain

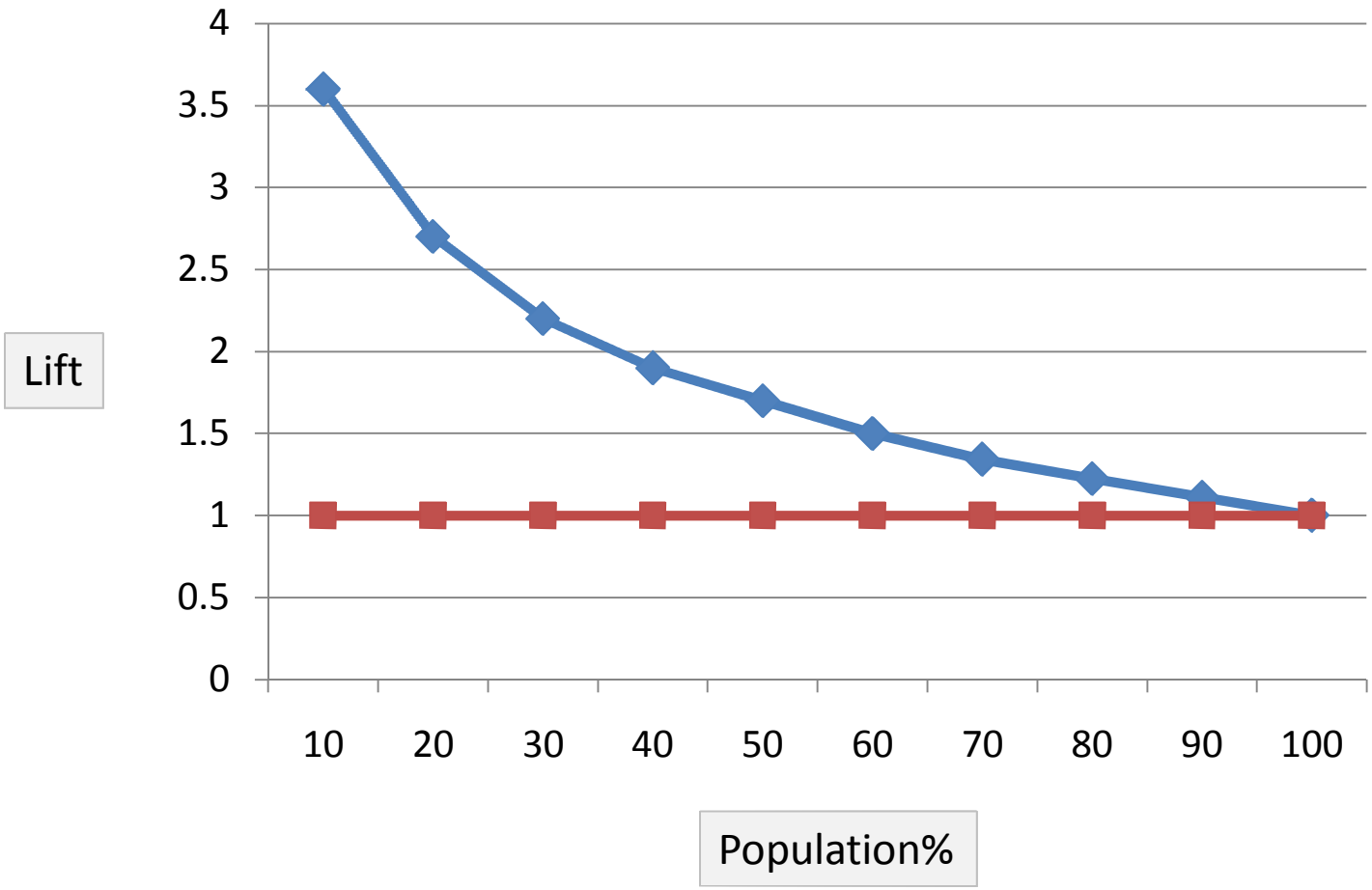
Population%	Target%
10	36
20	54
30	66
40	76
50	85
60	90
70	94
80	98
90	100
100	100

36/10

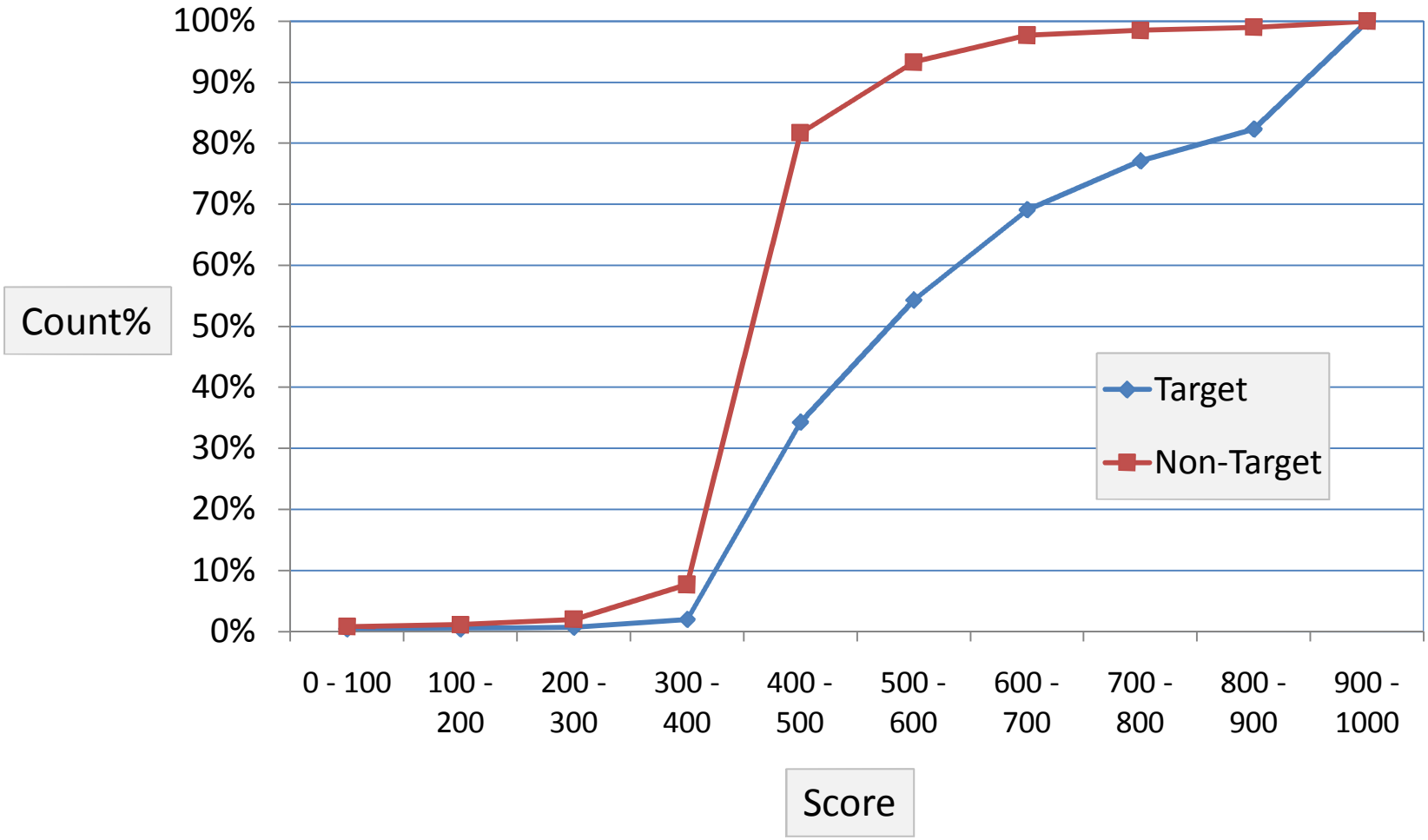
Lift

Population%	Lift
10	3.6
20	2.7
30	2.2
40	1.9
50	1.7
60	1.5
70	1.3
80	1.2
90	1.1
100	1

Lift Chart



K-S Chart


100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Count%
Target
Non-Target
0 - 100
100 - 200
200 - 300
300 - 400
400 - 500
500 - 600
600 - 700
700 - 800
800 - 900
900 - 1000
Score

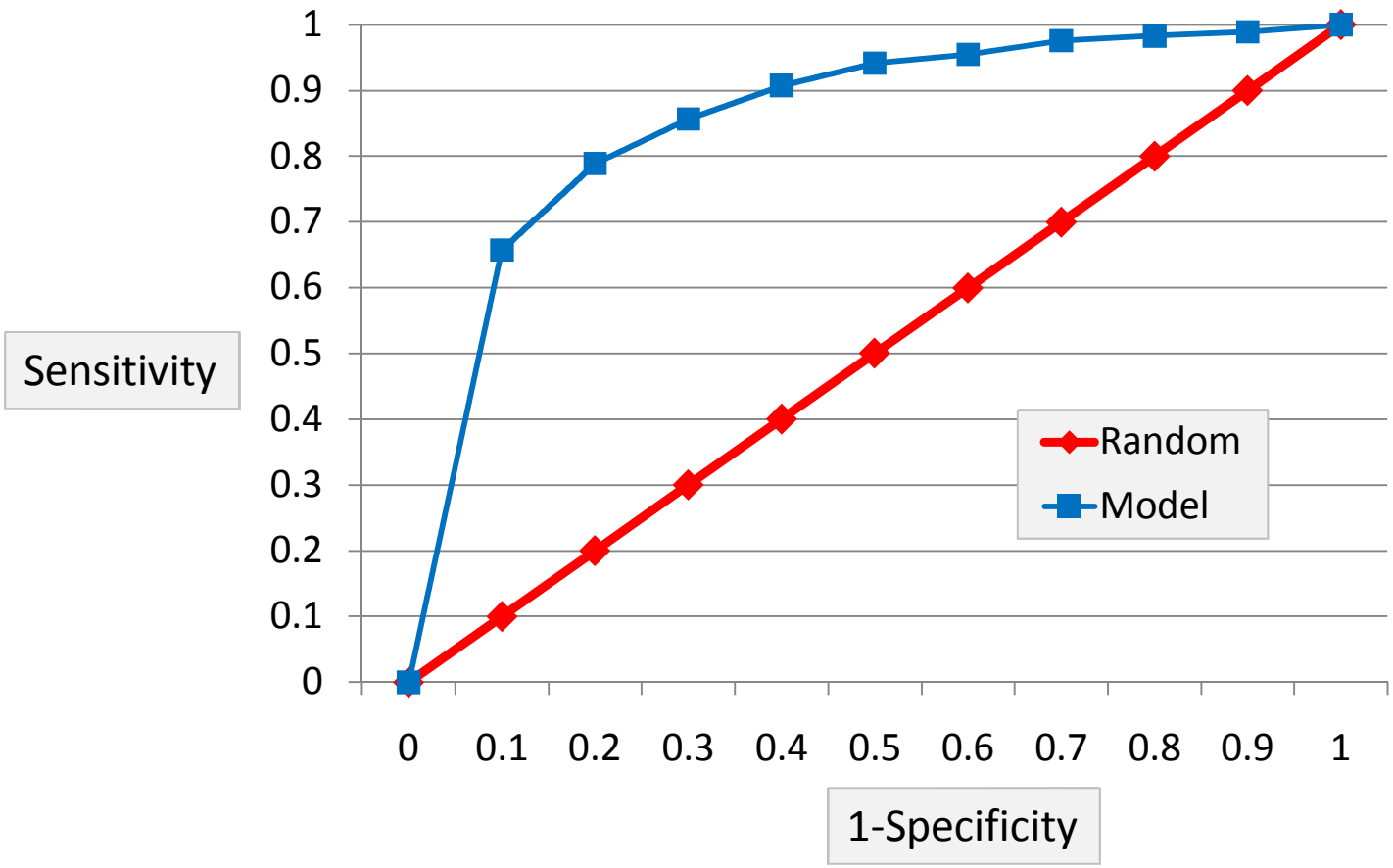
K-S Chart (Kolmogorov-Smirnov)

Score Range		Count		Cumulative Count		
Lower	Upper	Target	Non-Target	Target	Non-Target	K-S
0	100	3	62	0.5%	0.8%	0.3%
100	200	0	23	0.5%	1.1%	0.6%
200	300	1	66	0.7%	2.0%	1.3%
300	400	7	434	2.0%	7.7%	5.7%
400	500	181	5627	34.3%	81.7%	47.4%
500	600	112	886	54.3%	93.3%	39.0%
600	700	83	332	69.1%	97.7%	28.6%
700	800	45	63	77.1%	98.5%	21.4%
800	900	29	37	82.3%	99.0%	16.7%
900	1000	99	77	100.0%	100.0%	0.0%

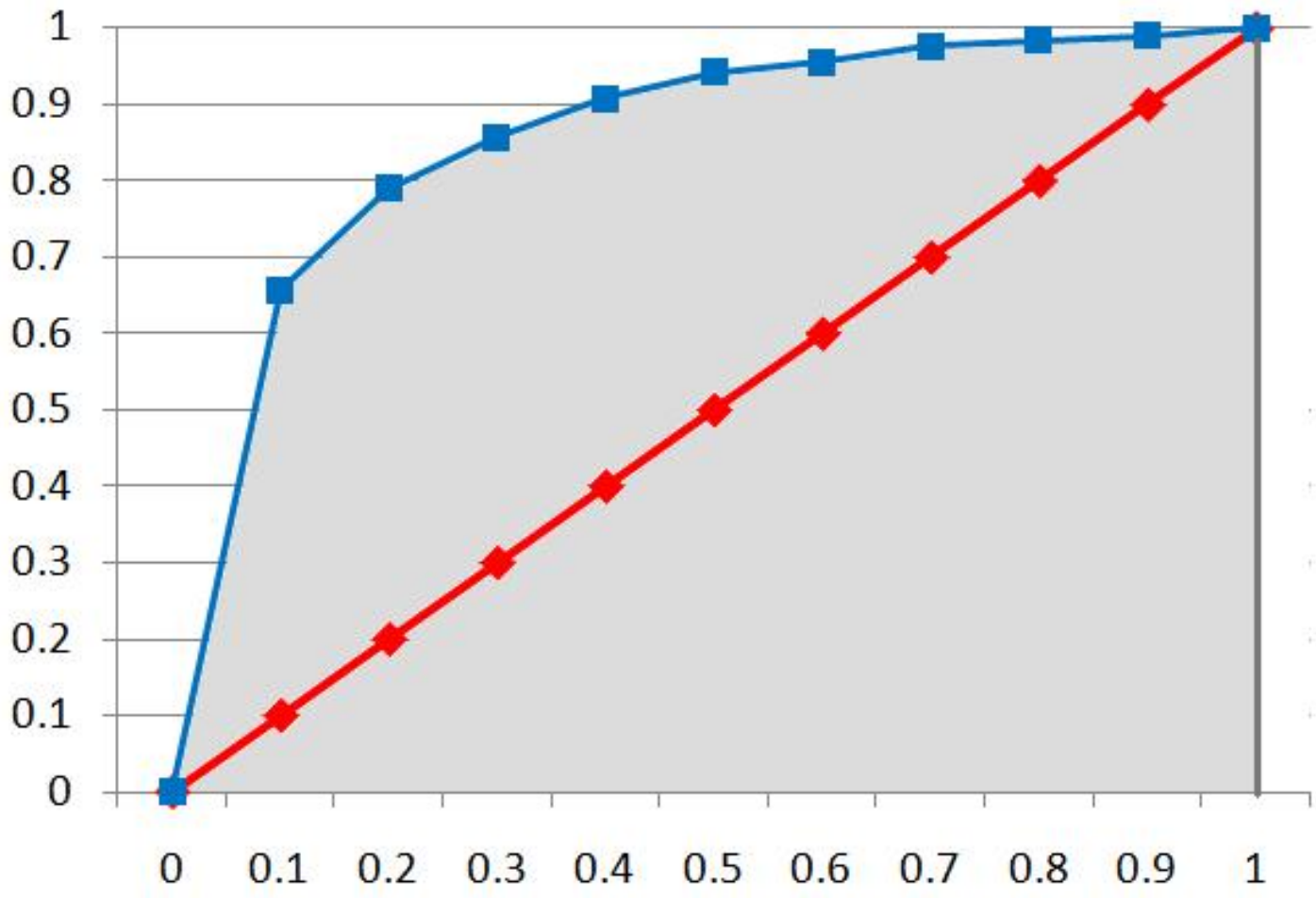
K-S

K(0.95) = 6.0%
K(0.99) = 7.1%

ROC Chart


Sensitivity
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1-Specificity
Random
Model

Area Under ROC Curve

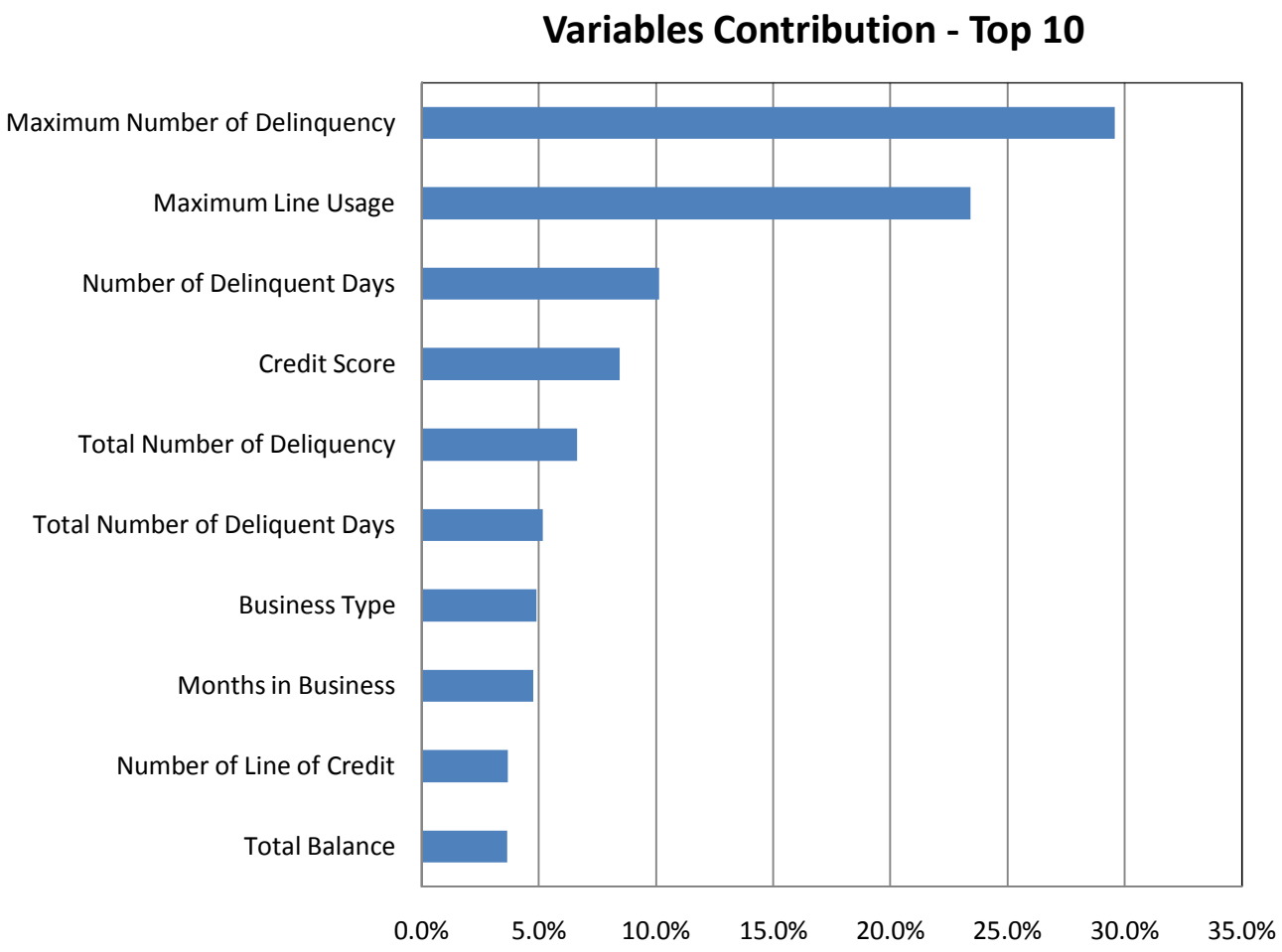


Model Comparison

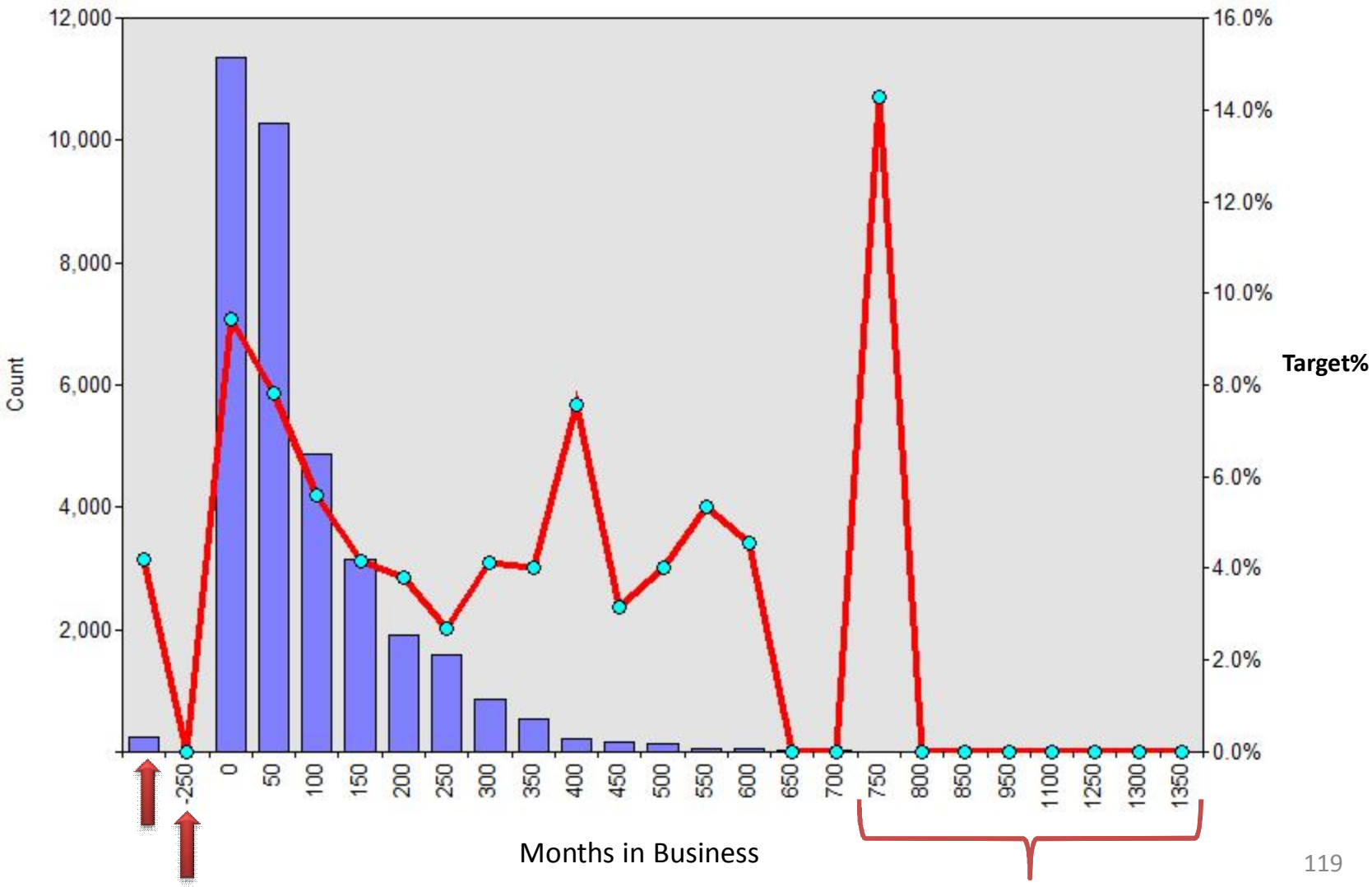
Model Comparison

Model	Accuracy	PPV	Sensitivity	ROC Area
ZeroR ✗	0.93	0.0	0.0	0.50
OneR ✗	0.94	0.23	0.62	0.61
Bayesian ✓	**0.93**	**0.36**	0.44	**0.79**
Decision Tree ✗	0.94	0.20	0.70	0.73
Logistic Regression ✓	**0.94**	0.24	0.73	**0.85**
LDA ✓	**0.93**	0.24	0.71	**0.83**
SVM (Lin) ✗	0.94	0.07	**0.83**	0.54
KNN (5NN) ✗	0.78	0.27	0.09	0.59
Neural Network (RBF) ✗	0.94	0.25	0.62	0.78

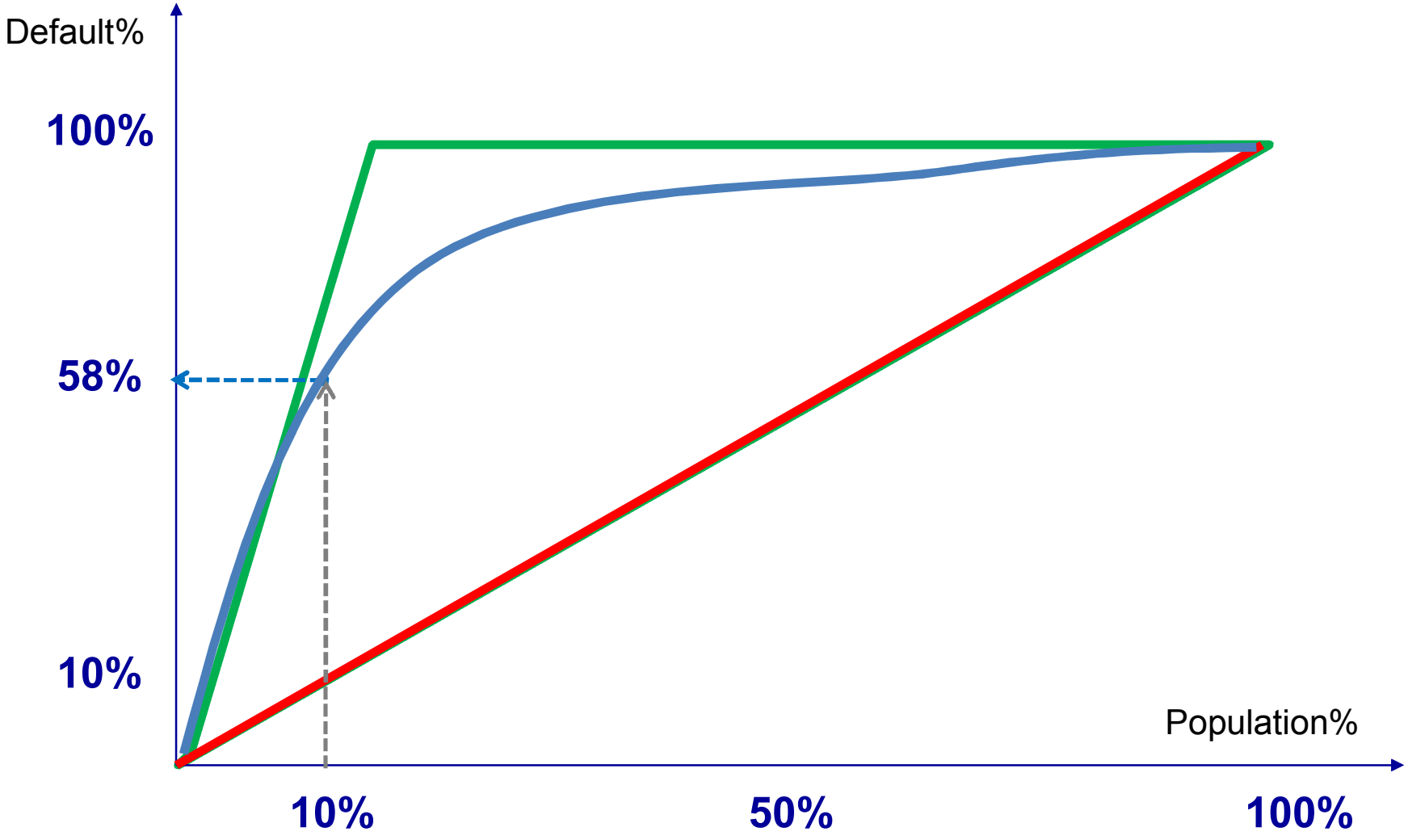
Evaluation – Variables Contribution



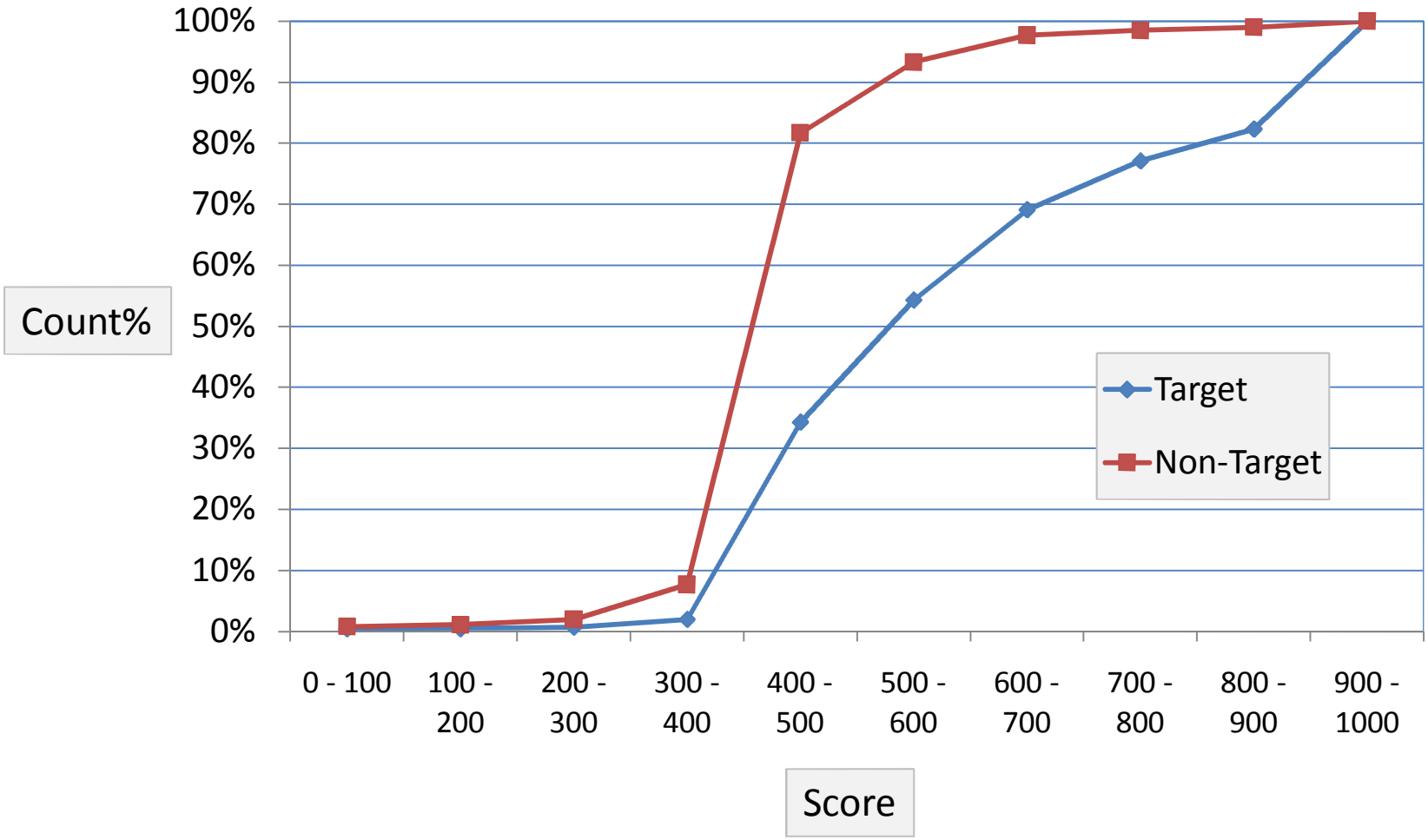
Data Exploration - Bivariate



Evaluation – Gain Chart



K-S Chart


100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Count%
Target
Non-Target
0 - 100
100 - 200
200 - 300
300 - 400
400 - 500
500 - 600
600 - 700
700 - 800
800 - 900
900 - 1000
Score

Final Result

- Total number of cases = **8,167**
- Total number of targets = **560**
- Total balance for targets = **$12,281,589**
- Top 10% **Random**
  - Number of targets = **56**
  - Total balance = **$1,230,000**
- Top 10% **Model**
  - Number of targets = **305**
  - Total balance = **$7,655,772**

600% Lift

The Amount of Balance?

Loss Given Default?

THANK YOU!

saed.sayad@ismartsoft.com

Prepayment Modeling (Refinance, Move, and Default)