A. JAMES CLARK SCHOOL OF ENGINEERING

ENPM808L

Analytics for Decision Support

Week 3

Introduction to Analytics and Decision Support

W. Lewis Collier, Ph.D.

Agenda

- Before class
  - Complete and turn in Python exercise 2
  - Read Chapter 4 : Fitting a Model to Data
- In class
  - Review and questions
  - Linear Discriminant Functions
  - Logistic Regression
  - Questions
  - Exercise discussion
- After class
  - Python exercise 3

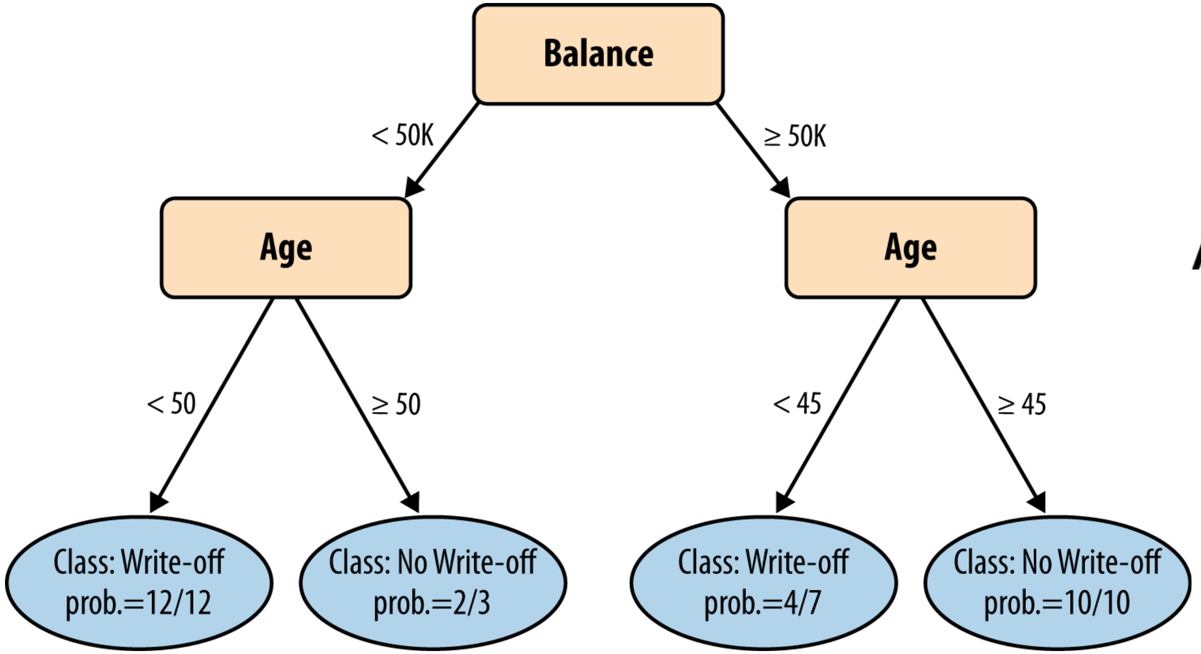
ENPM808L
Week 3

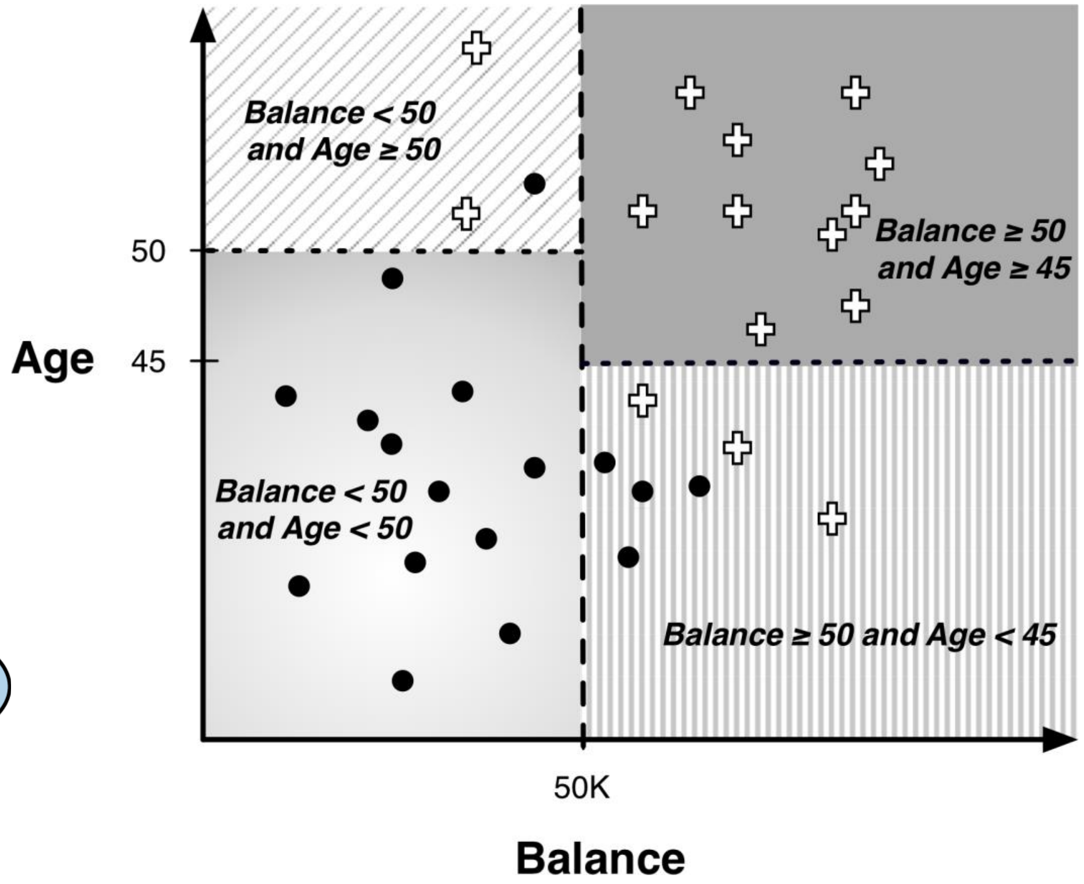
Review and Questions

ENPM808L
Week 3

Fitting a Model to Data

Multidimensional Decision Boundaries



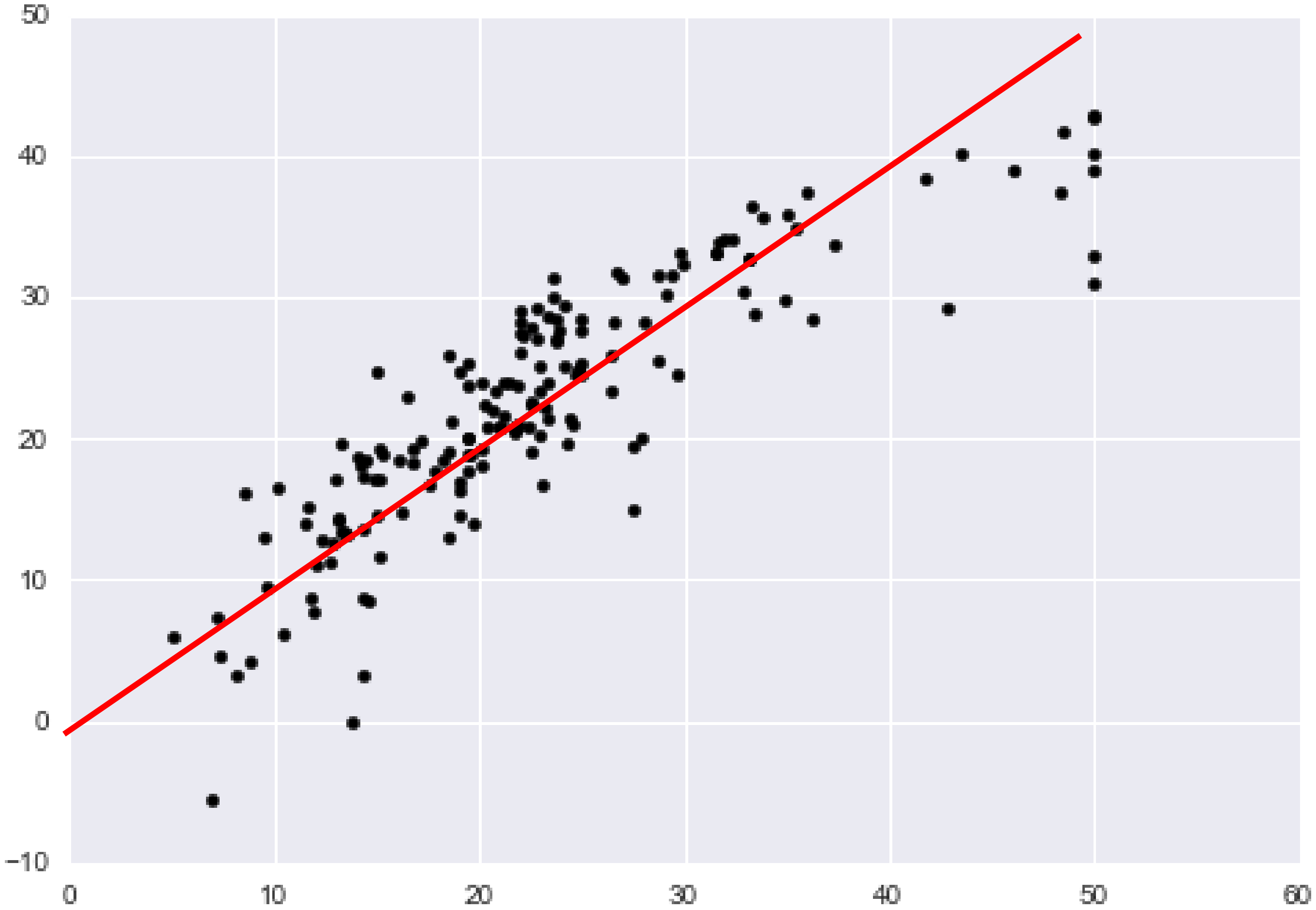


An Optimization Problem

- Fitting the model by
  - Calculating weights till an Objective (or Loss) Function is optimized (goal)
  - Three variations use different objective functions:
    - Linear regression
    - Support Vector Machines (SVM)
    - *Logistic regression*

Linear Classifiers

- Recall linear “fit” of data to create a function that allows for prediction of future values from a measurement
- This only works for simple X vs. Y data types
- We need a more complex way to specify data models with a mathematical basis



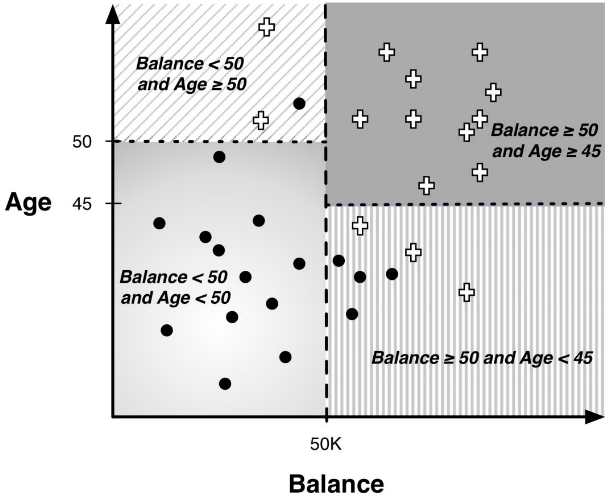
Linear Regression

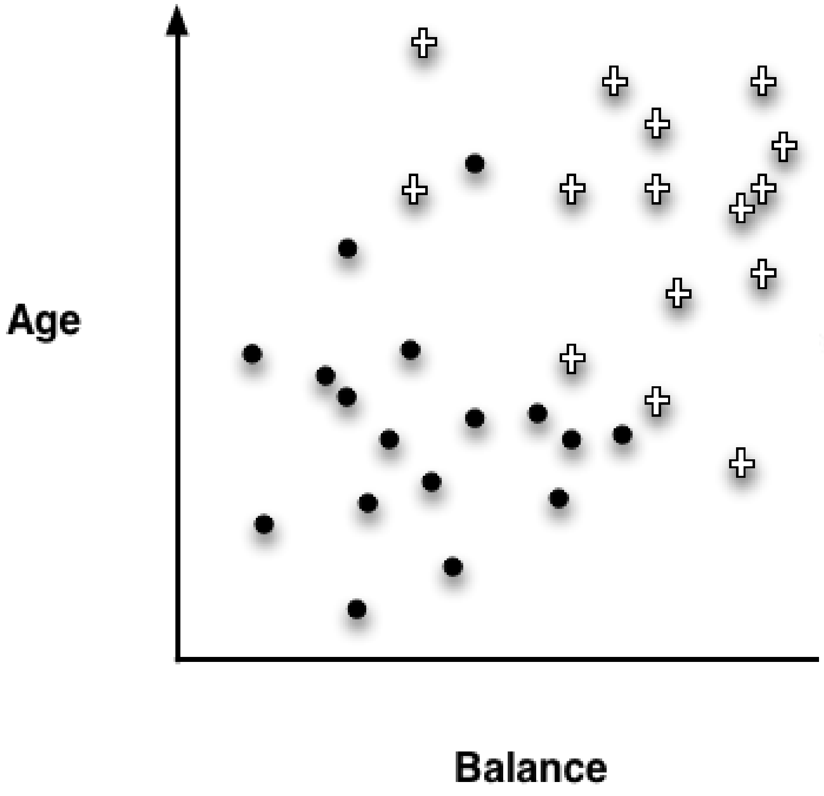
- A parametric model

$$f\,(\mathrm{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots$$

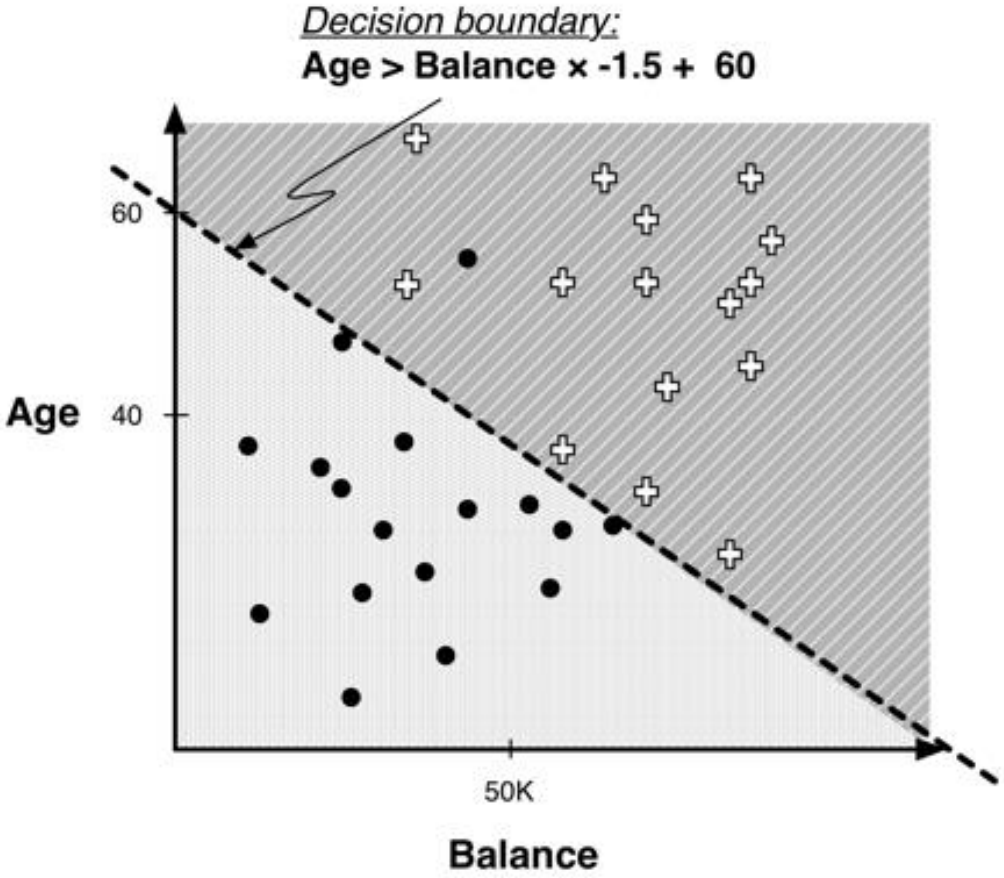
- Target variable is Numeric
- Objective Function:
  - Minimize sum of errors between true and estimated values
    - Absolute errors
    - Squared errors
- Only works on specific classes of problems.
- Can extend to Multidimensional Decision Boundaries (like decision trees)

Breaking Instance Space by Decision Boundaries





Instance Space

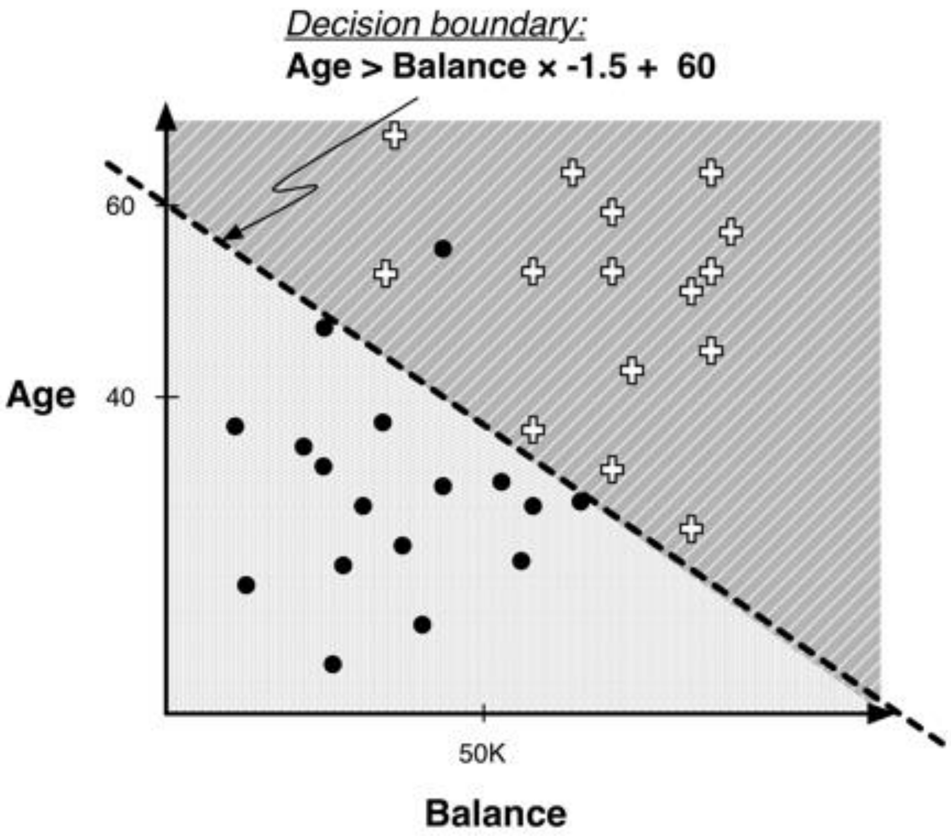


Decision Boundary with Linear Classifier

$y = mx + b$

Score and Rank Dataset Instances

- Use Linear Discriminant Functions
  - Measure "likeliness" to belong to a class
    - Which clients are most likely to buy a product?
    - Use a probability of membership in a class
    - Rank the probabilities



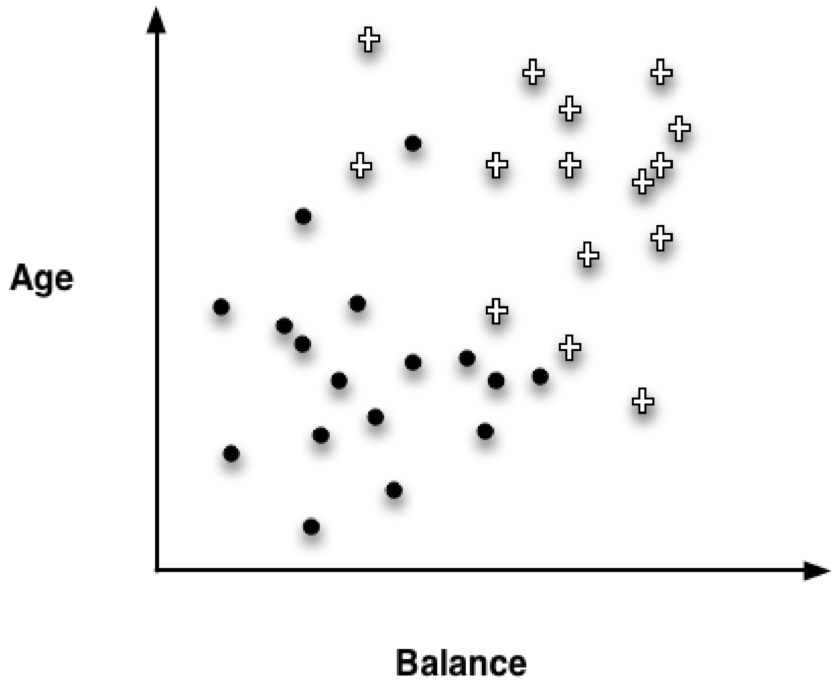
Linear Discriminant Model

- Create a model formed by a mathematical function of multiple attributes – (discriminating between classes)
- Classification Function
  - *class* =

    + if $1.0 \times Age - 1.5 \times Balance + 60 > 0$

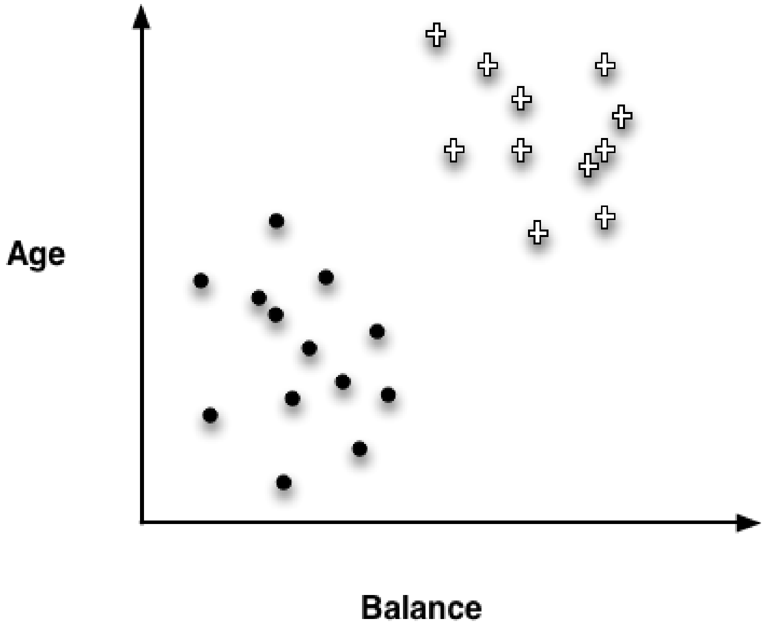
    • if $1.0 \times Age - 1.5 \times Balance + 60 \leq 0$
- General Linear Function:

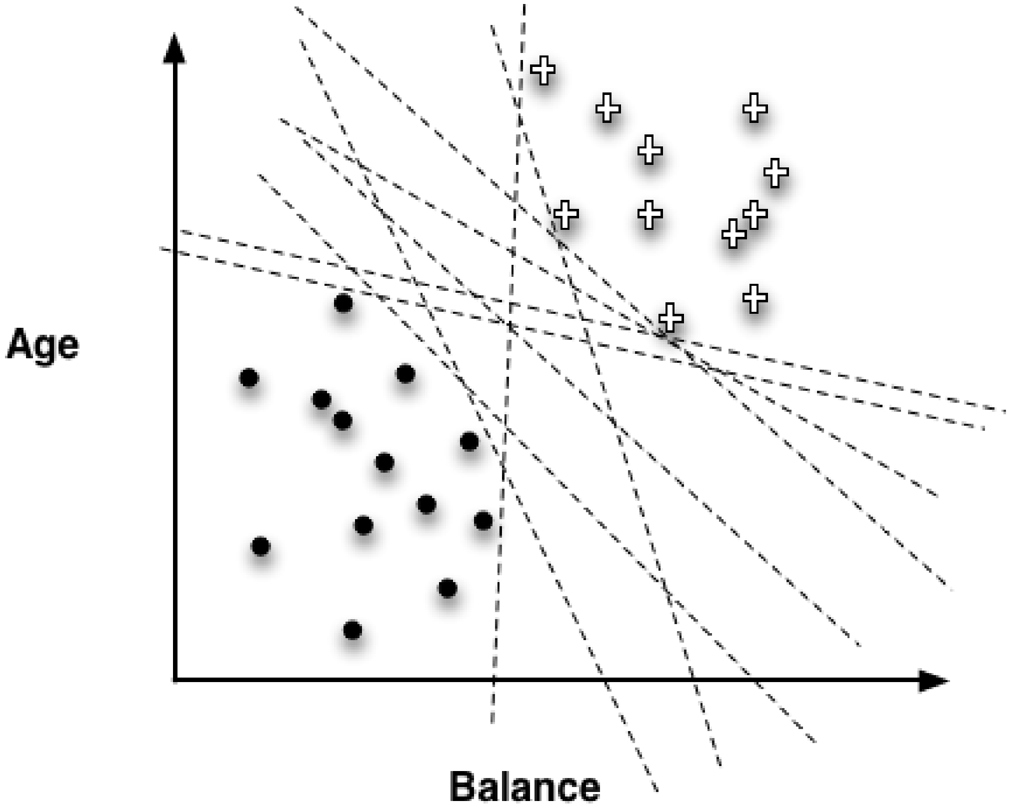
$$f(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots$$



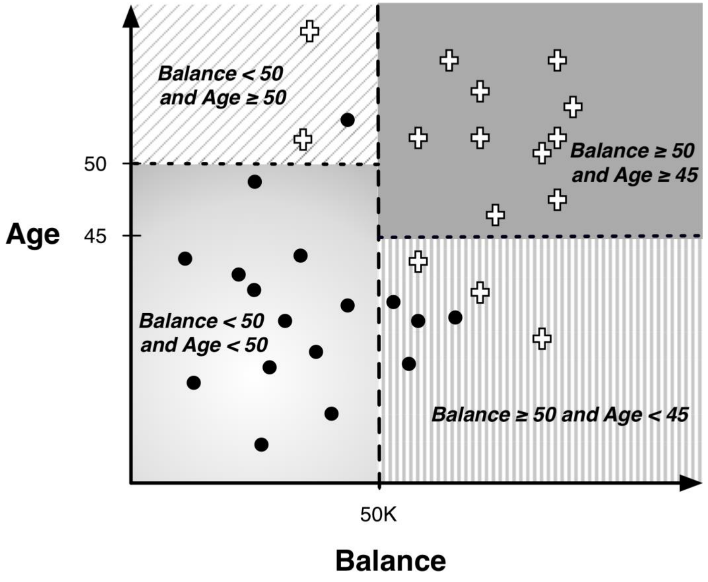
Linear Discriminant Model

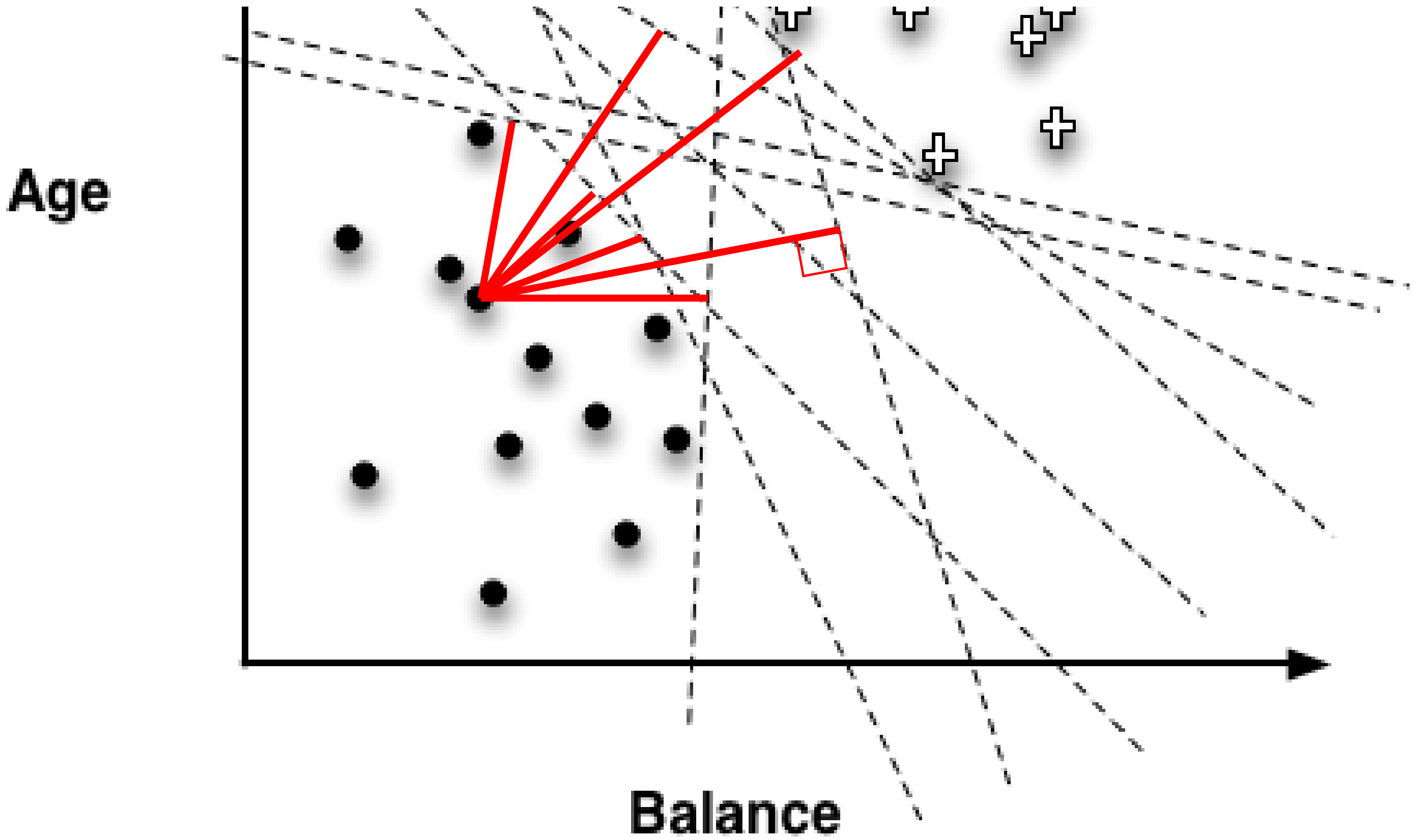
- Calculating weights (w) to fit a particular model – a parametrized model
- Finding best line to separate the classes





Linear Discriminant Model (details)

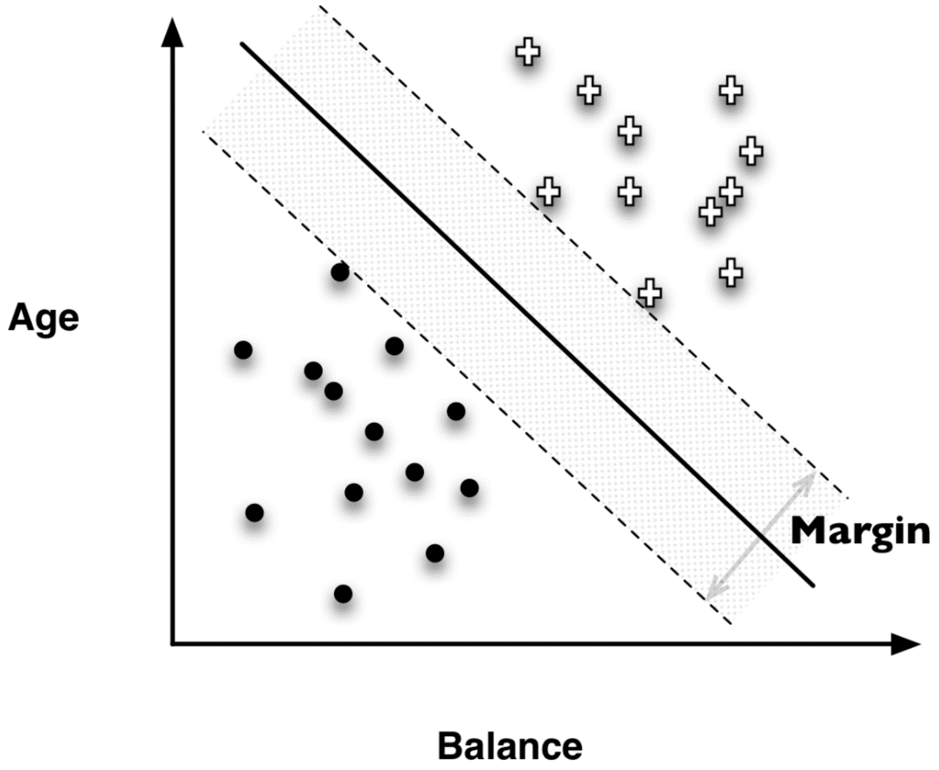




Support Vector Machines

- Linear Discriminant Model
  - Objective Function:
    - Maximize Margin

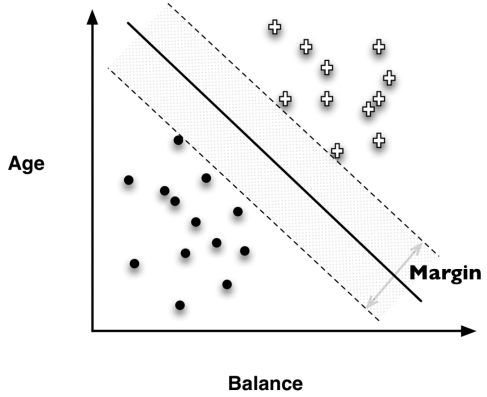
- What does this mean to the HW-2 problem of deciding what value to use as the “split value” between classes



Class Separation Decisions

	Num Samples																							
Setosa	50	0.528320834																						
Versicolor	50	0.528320834																						
Virginica	50	0.528320834																					Max IG	Split Value
Total	150						Top Segment								Bottom Segment								0.918295834	0.6
Entropy All	1.584962501	X[0]	X[1]	X[2]	X[3]		Setosa		Versicolor		Virginica				Setosa		Versicolor		Virginica					
		Sepal Length	Sepal Width	Petal Length	Petal Width	Class	#	Entopy	#	Entopy	#	Entopy	Total #	Entrophy	#	Entopy	#	Entopy	#	Entopy	Total #	Entrophy	IG	
		5	3.4	1.6	0.4	setosa	46	0	0	0	0	0	46	0	4	0.1808	50	0.50797285	50	0.508	104	1.196732	0.755228423	
		5.4	3.4	1.5	0.4	setosa	47	0	0	0	0	0	47	0	3	0.1486	50	0.506138028	50	0.5061	103	1.160865	0.787835515	
		5.1	3.8	1.9	0.4	setosa	48	0	0	0	0	0	48	0	2	0.1112	50	0.504200565	50	0.5042	102	1.119625	0.823617395	
		5.1	3.3	1.7	0.5	setosa	49	0	0	0	0	0	49	0	1	0.0659	50	0.502156086	50	0.5022	101	1.070235	0.864337562	
	2.45	5	3.5	1.6	0.6	setosa	50	0	0	0	0	0	50	0	0	0	50	0.5	50	0.5	100	1	0.918295834	0.6
		4.9	2.4	3.3	1	versicolor	50	0.028	1	0.1112	0	0	51	0.139233	0	0	49	0.502198909	50	0.4977	99	0.999926	0.877671857	
		5	2	3.5	1	versicolor	50	0.0544	2	0.1808	0	0	52	0.235193	0	0	48	0.504366046	50	0.4953	98	0.9997	0.85029176	
		6	2.2	4	1	versicolor	50	0.0793	3	0.2345	0	0	53	0.313813	0	0	47	0.506497191	50	0.4928	97	0.99931	0.827861519	
		5.8	2.7	4.1	1	versicolor	50	0.1028	4	0.2781	0	0	54	0.380947	0	0	46	0.508587761	50	0.4902	96	0.998747	0.808623459	
		5.7	2.6	3.5	1	versicolor	50	0.125	5	0.3145	0	0	55	0.439497	0	0	45	0.510632769	50	0.4874	95	0.998001	0.791746379	
		5.5	2.4	3.7	1	versicolor	50	0.146	6	0.3453	0	0	56	0.491237	0	0	44	0.51262679	50	0.4844	94	0.997059	0.776743551	

- Walking through the data, looking for the peak IG gives rise to a split value of 1.6
- Taking the 1st point after the peak gives a split value of 3.3
- Better approach is to take the average, which gives 2.45

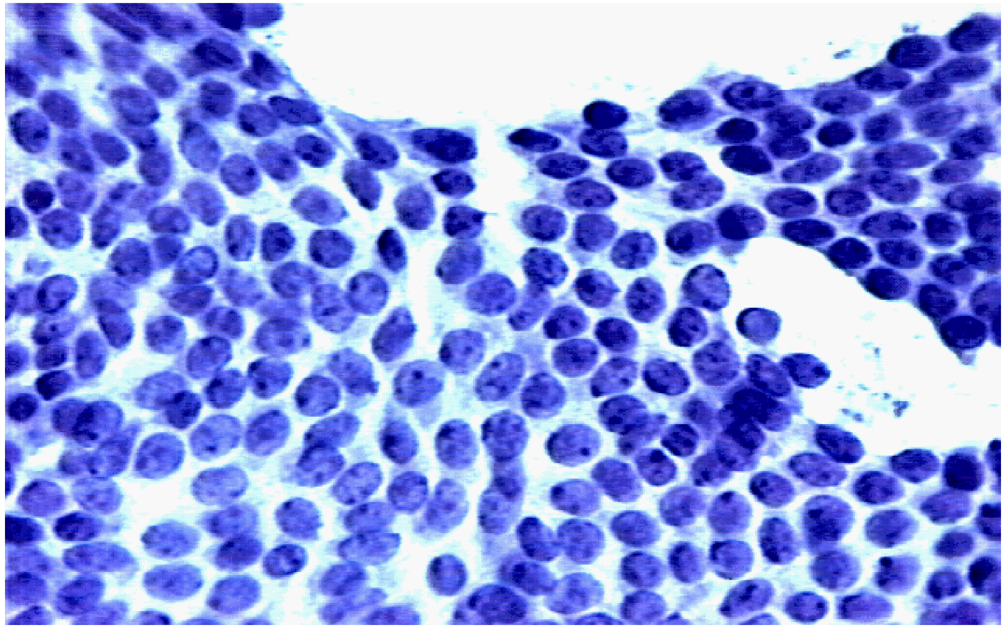


Discussion

- Characteristics of the Decision/Classification Tree Model vs. Linear Classifiers

Wisconsin Breast Cancer Dataset

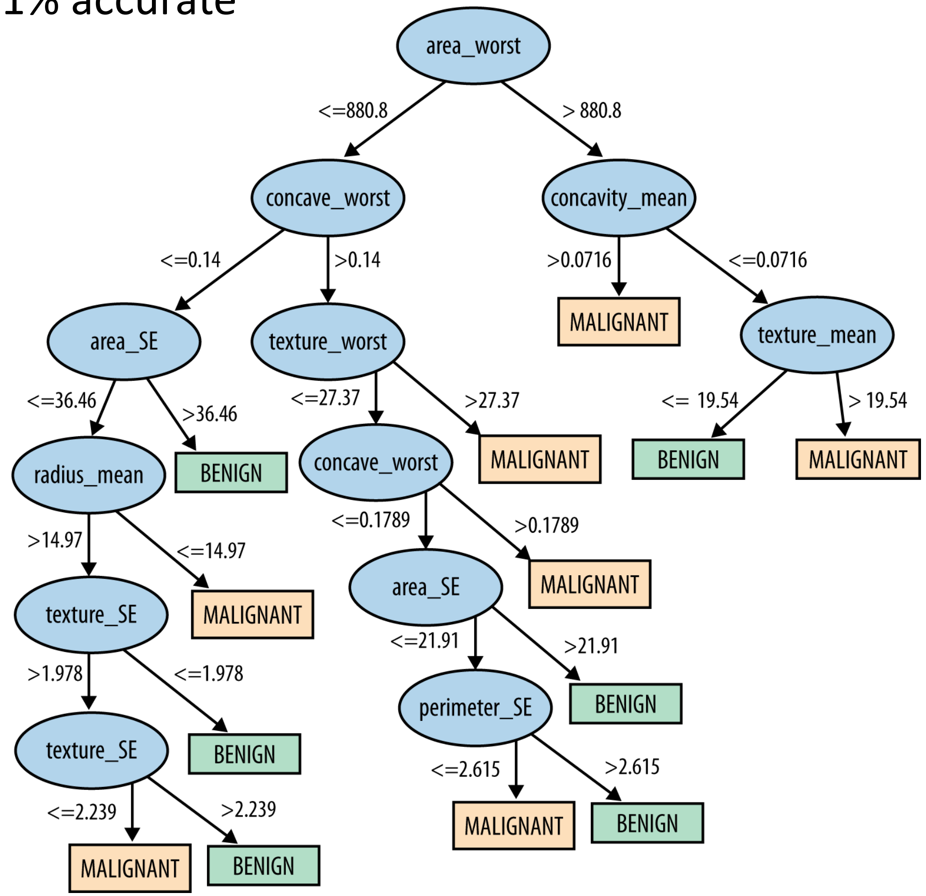
- Predictive model: Identify Malignant or Benign cell nuclei images



Attribute name	Description
RADIUS	*Mean of distances from center to points on the perimeter*
TEXTURE	*Standard deviation of grayscale values*
PERIMETER	*Perimeter of the mass*
AREA	*Area of the mass*
SMOOTHNESS	*Local variation in radius lengths*
COMPACTNESS	*Computed as:* $perimeter^2/area - 1.0$
CONCAVITY	*Severity of concave portions of the contour*
CONCAVE POINTS	*Number of concave portions of the contour*
SYMMETRY	*A measure of the nucleii's symmetry*
FRACTAL DIMENSION	*'Coastline approximation'* – 1.0
DIAGNOSIS (Target)	*Diagnosis of cell sample: malignant or benign*

Compare Results/Accuracy of two models

99.1% accurate



Attribute	Weight (learned parameter)
SMOOTHNESS_worst	22.3
CONCAVE_mean	19.47
CONCAVE_worst	11.68
SYMMETRY_worst	4.99
CONCAVITY_worst	2.86
CONCAVITY_mean	2.34
RADIUS_worst	0.25
TEXTURE_worst	0.13
AREA_SE	0.06
TEXTURE_mean	0.03
TEXTURE_SE	−0.29
COMPACTNESS_mean	−7.1
COMPACTNESS_SE	−27.87
w_0 *(intercept)*	−17.7

98.9% accurate

Case Study - Flight Delay Dataset

- Predicting if a flight will be delayed

DAY_OF_WEEK	CARRIER	ORIGIN	DEST	DEP_TIME_BLK	DELAY_WEATHER	DELAY_15
1	CO	DCA	EWR	1700-1759	0	0
1	CO	DCA	EWR	1300-1359	0	1
1	CO	DCA	EWR	0700-0759	0	0
1	CO	DCA	EWR	0700-0759	1	1
1	CO	DCA	EWR	1900-1959	0	1
1	CO	DCA	EWR	1300-1359	0	0
1	CO	DCA	EWR	1300-1359	0	0
1	CO	DCA	EWR	0700-0759	0	0
1	CO	DCA	EWR	0700-0759	0	0
1	CO	DCA	EWR	1300-1359	0	0
1	CO	DCA	EWR	1700-1759	0	1
1	CO	DCA	EWR	1700-1759	0	0
1	CO	DCA	EWR	1900-1959	0	0
1	CO	DCA	EWR	1700-1759	0	0
1	CO	DCA	EWR	1900-1959	0	0
1	DH	DCA	JFK	1600-1659	0	0
1	DH	IAD	LGA	1200-1259	0	1
1	DH	IAD	LGA	1400-1459	0	1
1	DH	IAD	LGA	1700-1759	0	1
1	DH	IAD	EWR	0800-0859	0	0
1	DH	IAD	EWR	1700-1759	0	1
1	DH	IAD	EWR	1200-1259	0	1
1	DH	IAD	EWR	2100-2159	0	1
1	DH	IAD	EWR	1400-1459	0	1

http://stat-computing.org/dataexpo/2009/the-data.html

Flight Delay Dataset Characteristics

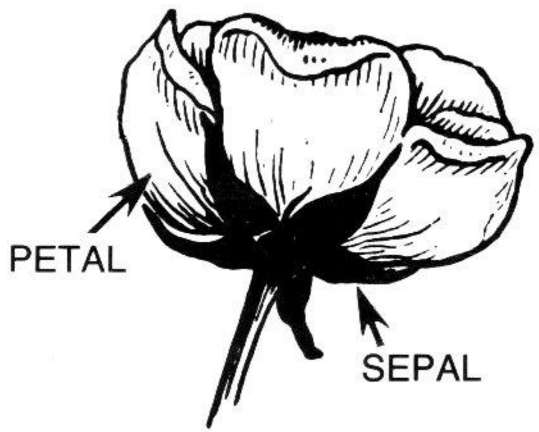
- DAY_OF_WEEK: Monday-Friday (1-7)
- CARRIER
- ORIGIN: BWI, DCA, IAD
- DEST: EWR, JFK, LGA
- DEP_TIME_BLK: Hourly blocks during day
- DELAY_WEATHER: 1=yes
- DELAY_15: Flight delayed by at least 15 min

Flight Delay Dataset

- Objectives
- Data Understanding and Processing
- Modeling
- Evaluation

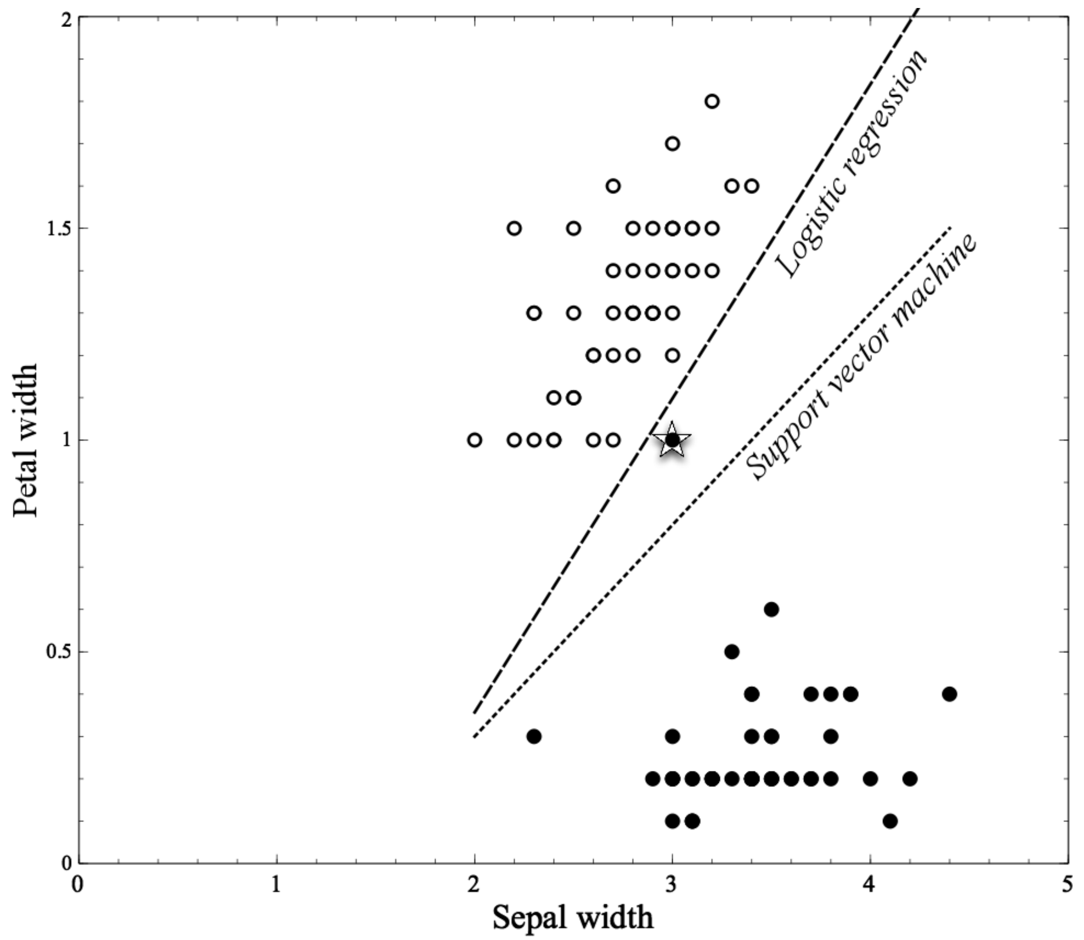
Building a Linear Discriminant from Data

- Example Iris Dataset (from UCI Data Repository):
  - Problem: Classify dataset into 3 species of flowering plants

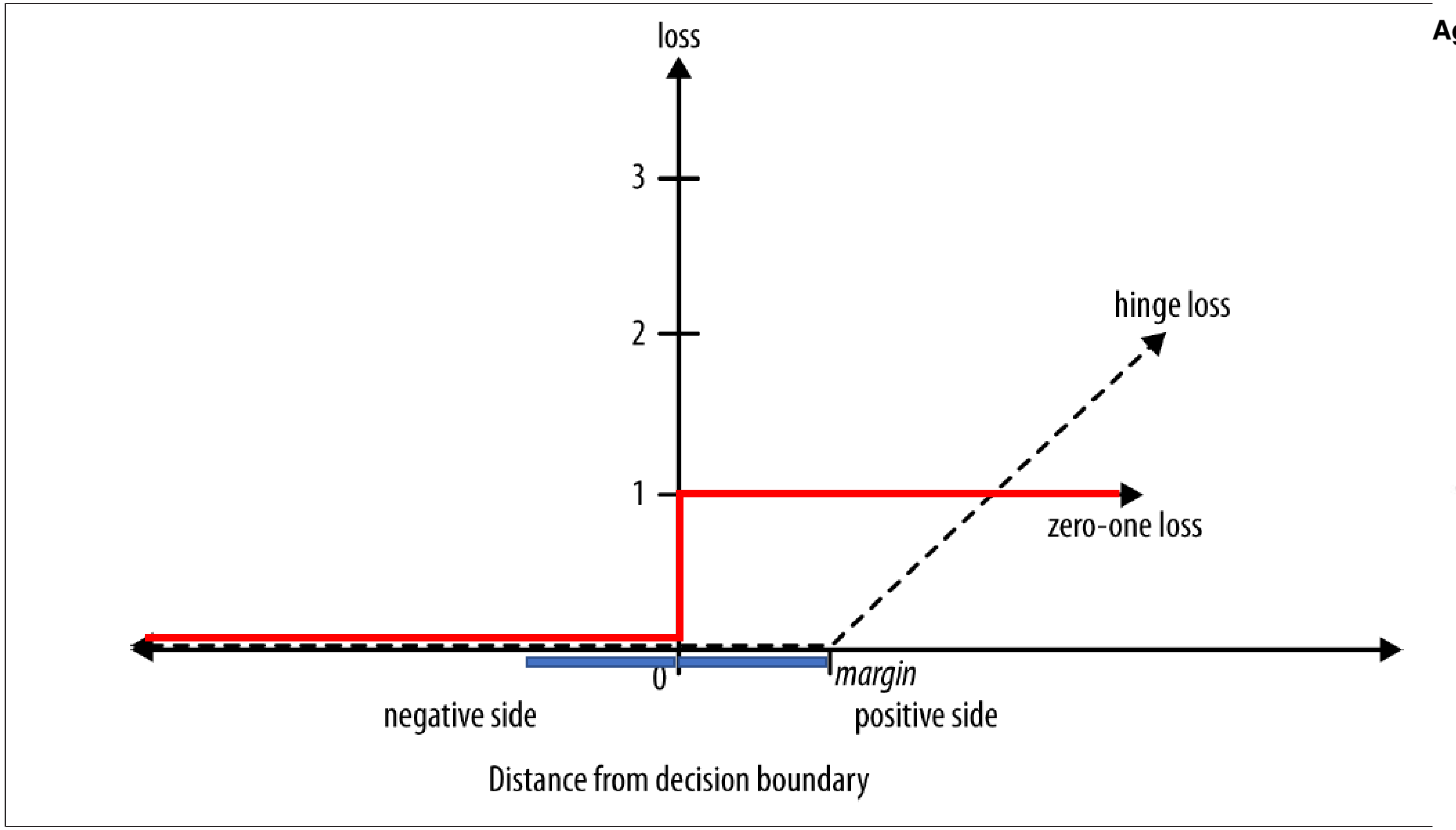


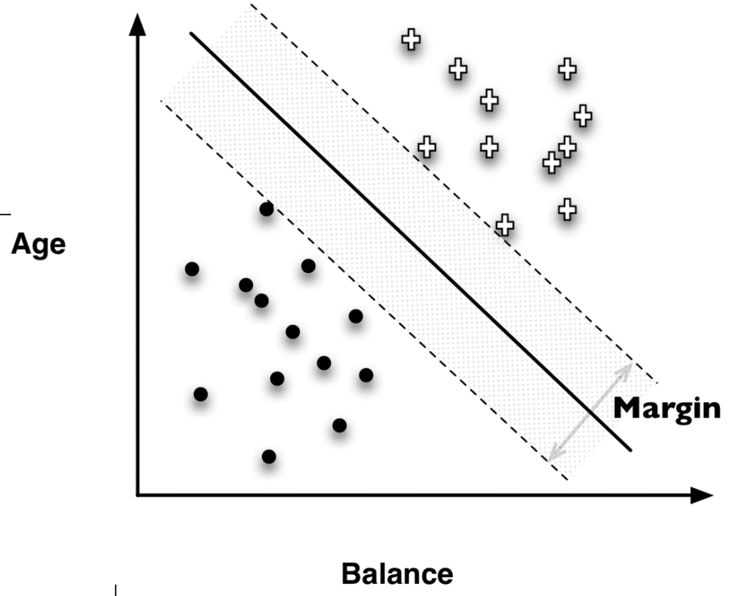
https://archive.ics.uci.edu/ml/machine-learning-databases/iris/

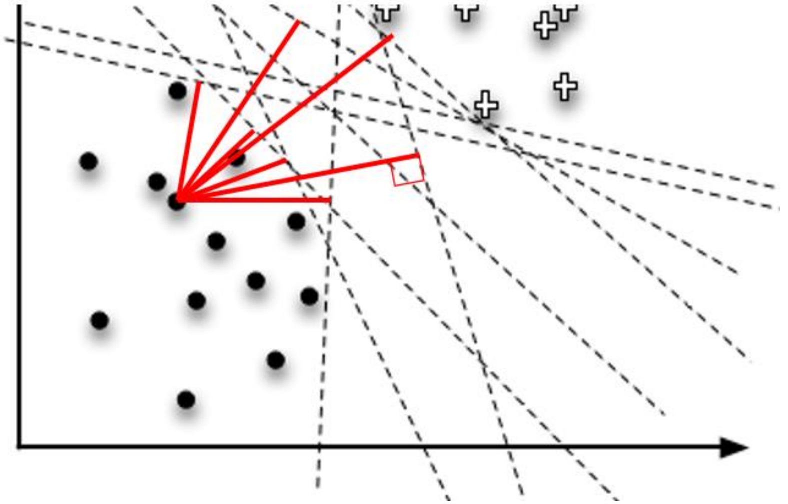
Building Linear Classifiers from Iris Class Data



Penalty functions

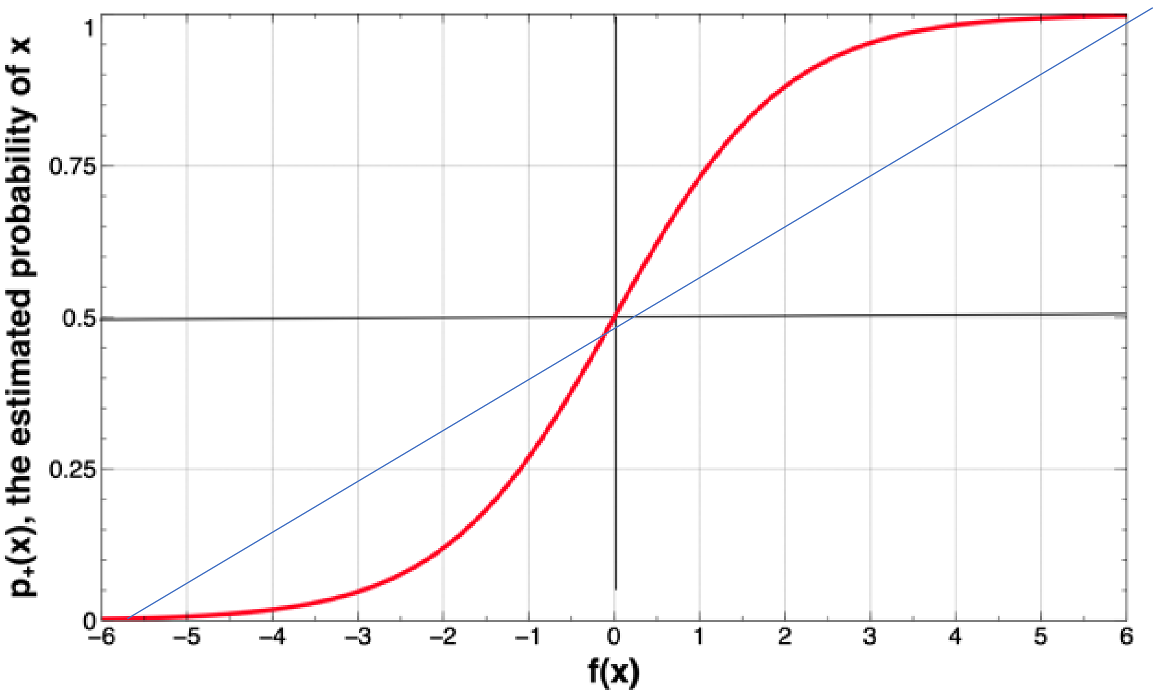






Logistic Regression

- In Logistic Regression, we don't directly fit a straight line to our data like in linear regression. Instead, we fit a S shaped curve, called ***Sigmoid***, to our observations.



Distance from Boundary

Logistic Regression Discussions

- https://towardsdatascience.com/logistic-regression-explained-9ee73cede081
- https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression
- https://aws.amazon.com/what-is/logistic-regression/
- https://en.wikipedia.org/wiki/Logistic_regression

What is Logistic Regression (LR)?

- In the Machine Learning world, **Logistic Regression** is a kind of **parametric classification model,** despite having the word '*regression*' in its name.
- LR models are models that have a certain fixed number of parameters that depend on the number of input features, and they output categorical prediction, like for example if a plant belongs to a certain species or not.
- In Logistic Regression, we don't directly fit a straight line to our data like in linear regression. Instead, we fit a S shaped curve, called ***Sigmoid***, to our observations.
- Simple LR provides a simple YES/NO answer, some versions can provide a 1 of N answer.

Why do we need LR?

- Outliers skew linear models, even with penalty functions
- We often just need a yes/no answer based on multiple parameters
  - Will a flight be delayed?
  - Is a given cell cancerous?
  - Is the flower a given species?
- The model is simple, so its effects can be better understood
- They can be fast
- They can be used to decompose a problem into more manageable pieces

Basic LR assumptions

- **The dependent variable is binary or dichotomous**—i.e. It fits into 1 of N clear-cut categories.
- **There should be no, or very little, multicollinearity between the predictor variables**—in other words, the predictor variables (or the independent variables) should be independent of each other.
- **The independent variables should be linearly related to the log odds.** **See https://careerfoundry.com/en/blog/data-analytics/what-is-logistic-regression/**
- **Logistic regression requires fairly large sample sizes**—the larger the sample size, the more reliable (and powerful) you can expect the results of your analysis to be.

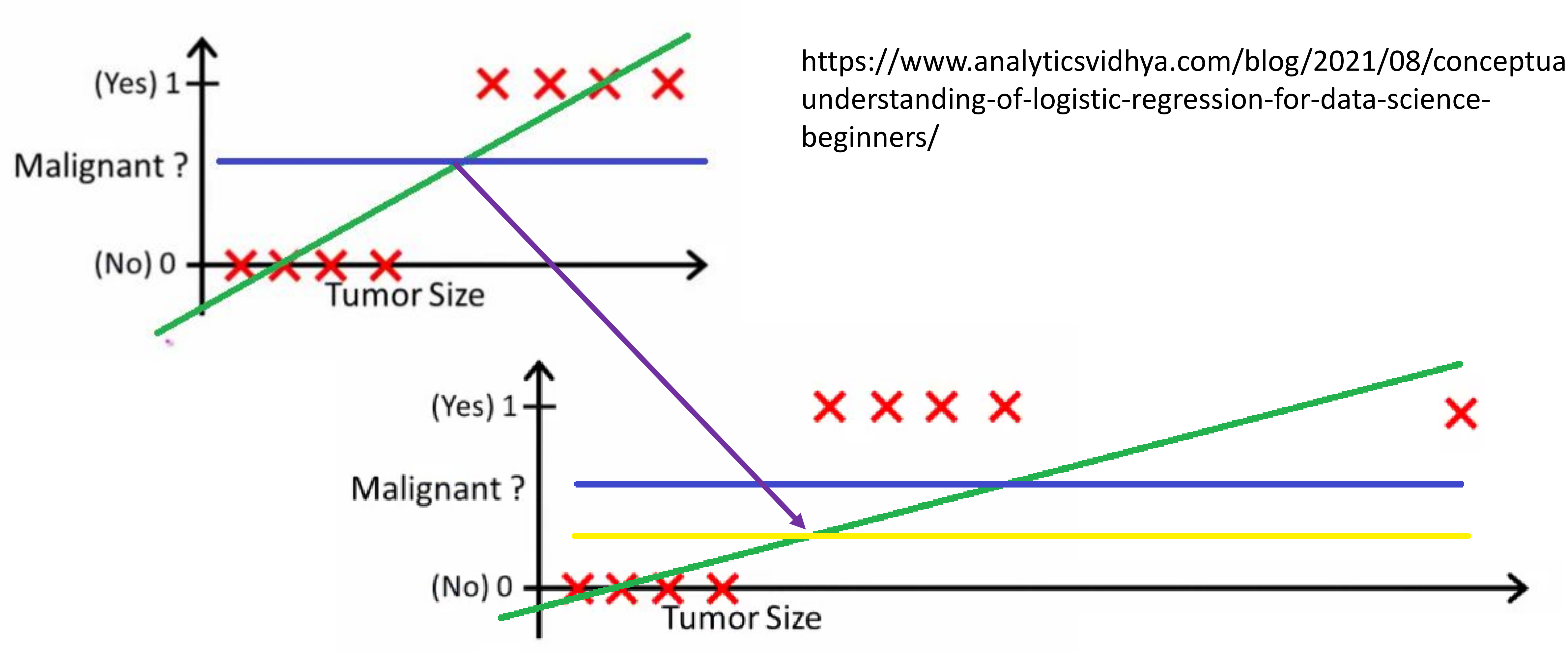
LR types

- **Binary logistic regression** is the statistical technique used to predict the relationship between the dependent variable (Y) and the independent variable (X), where the dependent variable is binary in nature.
- **Multinomial logistic regression** is used when you have one categorical dependent variable with two or more unordered levels (i.e., two or more discrete outcomes). It is very similar to logistic regression except that here you can have more than two possible outcomes.
- **Ordinal logistic regression** is used when the dependent variable (Y) is ordered (i.e., ordinal). The dependent variable has a meaningful order and more than two categories or levels. (agree/neutral/disagree)

How do LR algorithms work

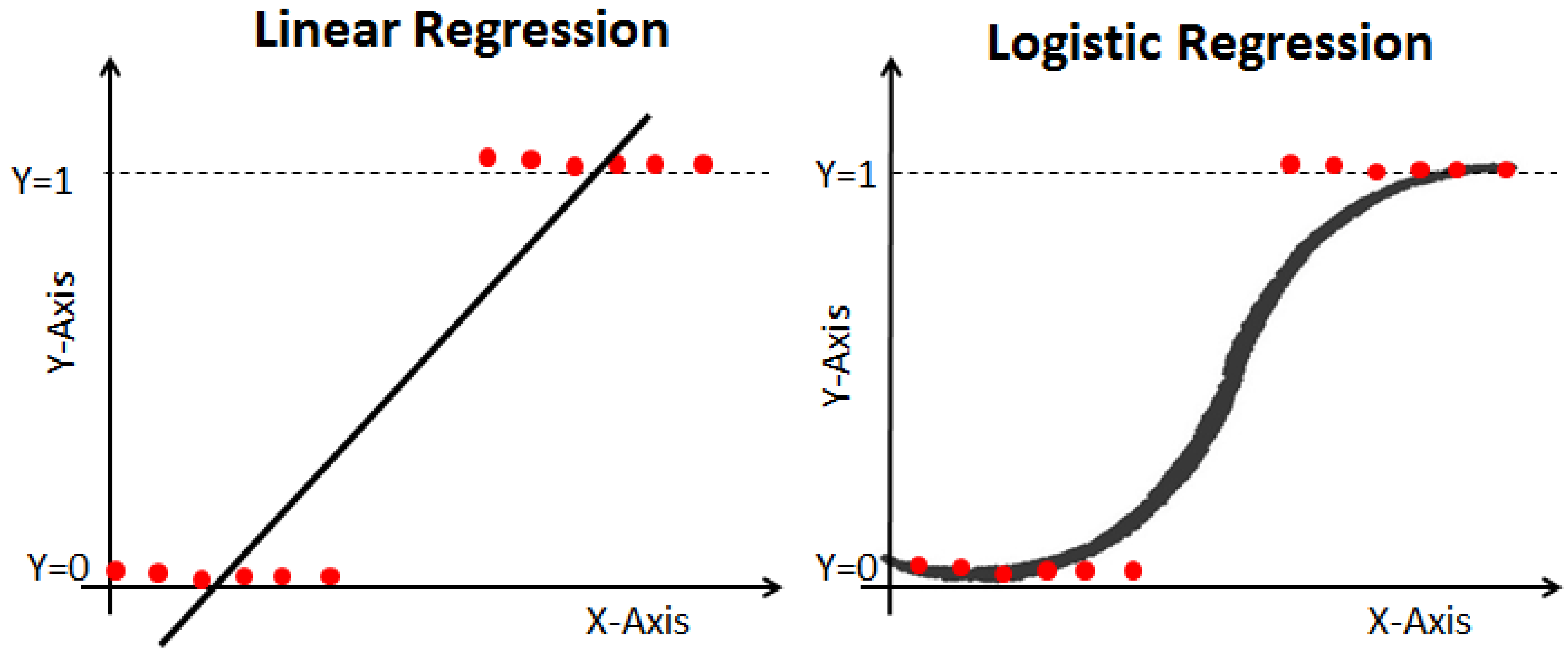
- Identify the question
  - Do rainy days impact our monthly sales? (yes or no)
- Collect historical data
  - collect the number of rainy days and your monthly sales data for each month in the past three years
- Train the LR analysis model
  - See the HW at end of these slides ☺
- Make predictions for unknown values
  - Use the software developed for the HW

Issues with Linear Analysis



https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

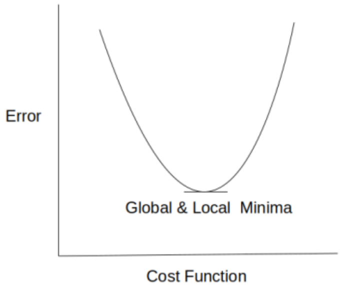
Linear vs Logistic



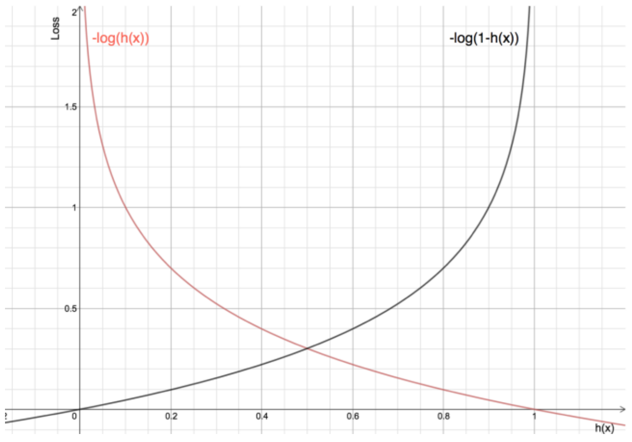
https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

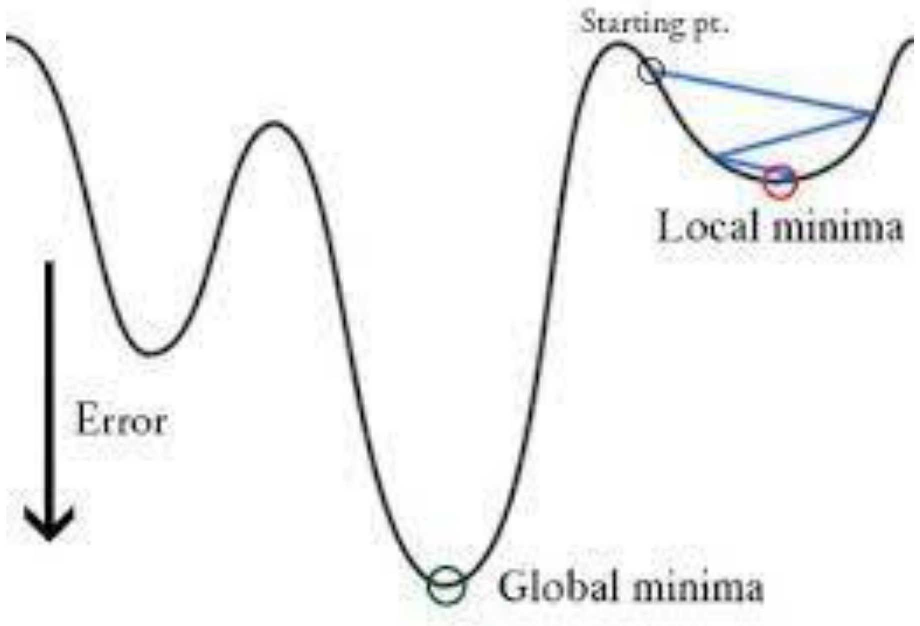
Optimization Issues with LR

- Linear regression cost function only has a single minima



- Use log loss function as the cost function
- Combine two-sided functions to get a single minima



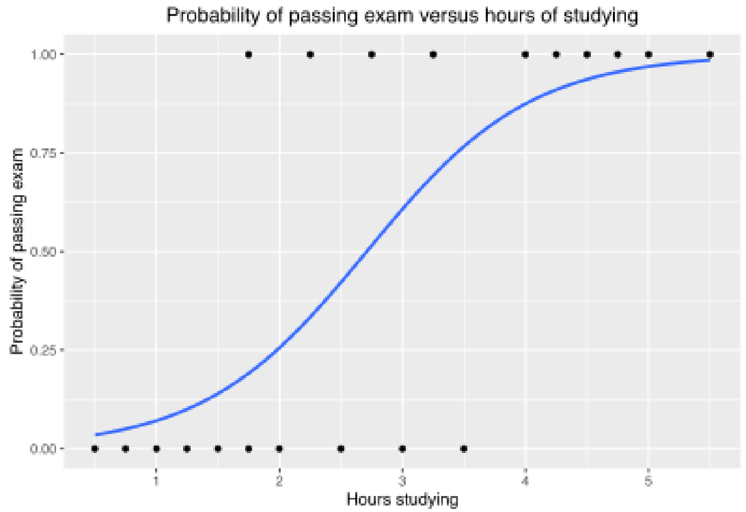


$$\text{Cost}(h_\Theta(x), y) = \begin{cases} -log(h_\Theta(x)) & if \quad y = 1 \\ -log(1 - h_\Theta(x)) & if \quad y = 0 \end{cases}$$

Another LR example

https://en.wikipedia.org/wiki/Logistic_regression#Example

- A group of 20 students spends between 0 and 6 hours studying for an exam. How does the number of hours spent studying affect the probability of the student passing the exam?
- If scores were used, then linear regression could be used, but with PASS/FAIL as results, need LR
- The usual measure of goodness of fit for a logistic regression uses logistic loss (or log loss), the negative log-likelihood.



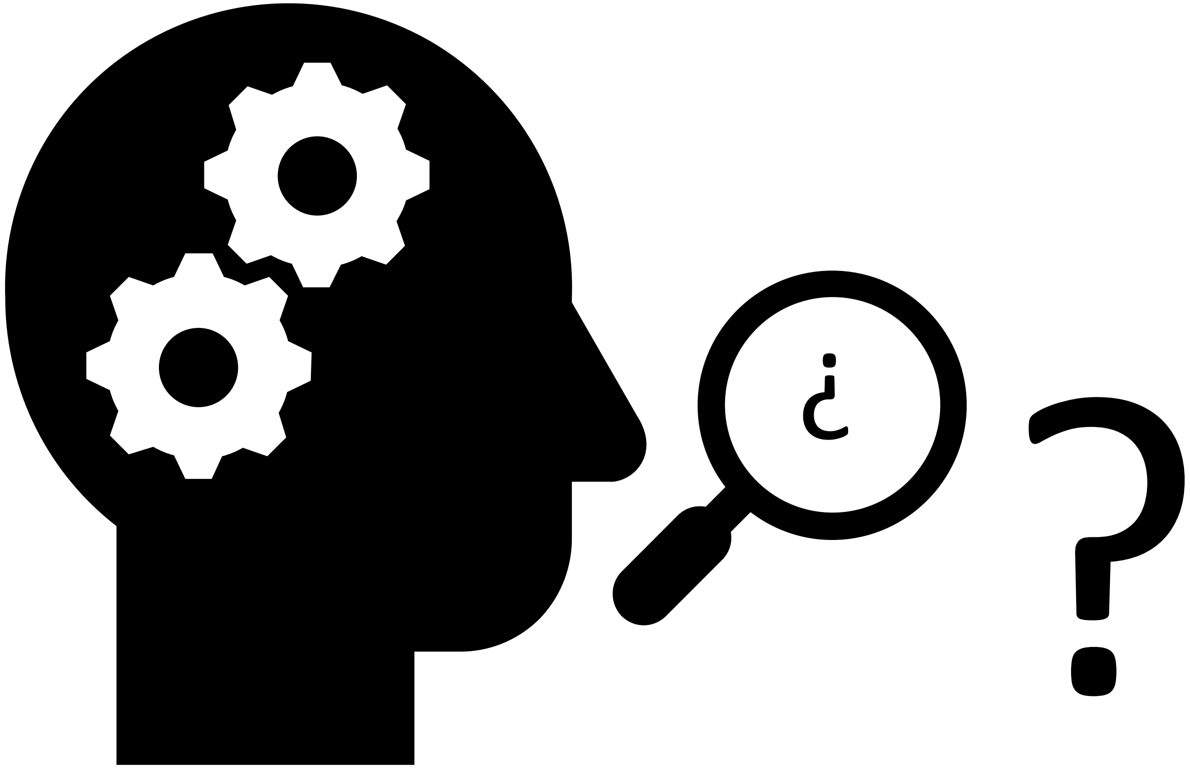
Hours of study (x)	Passing exam		
	Log-odds (t)	Odds (e^t)	Probability (p)
1	−2.57	0.076 ≈ 1:13.1	0.07
2	−1.07	0.34 ≈ 1:2.91	0.26
$\mu \approx 2.7$	0	1	$\frac{1}{2}$ = 0.50
3	0.44	1.55	0.61
4	1.94	6.96	0.87
5	3.45	31.4	0.97

Classification Tree vs. Logistic Regression

- Classification Tree
  - Selects single attribute at a time
  - Piecewise classifier using divide-and-conquer approach
  - Breaks instance space into smaller regions
- Linear Classifiers
  - Uses weighted combination of all attributes
  - Creates a single decision surface
- Logistic Regression
  - Uses Sigmoid curve instead of a line
  - Used simplified log loss as cost function

ENPM808L
Week 3

Questions



ENPM808L
Week 3

Exercise Discussion

HW-03 LR example

- Build LR code
  - from sklearn import linear_model
  - logr_model = linear_model.LogisticRegression()
- Test against Wiki pass/fail example data
- Test against Iris Data
- Present results in a report where:
  - Wiki example is shown as verifying your code, that is, the simple Wiki pass/fail example is used to show the code works
  - Iris data is the data described in the report, that is, the report discusses the Iris classification as the main focus.

sklearn Logistic Regression Example

- Build a Logistic Regression model for the training data from https://en.wikipedia.org/wiki/Logistic_regression#Example
- Compare the results with the answers provided in Wiki

sklearn Logistic Regression Example

- Use the iris dataset used for the previous example
- Run your Logistic Regression model for the same Iris training data
- Compare the results with the previous decision tree model
- Create a ranking of the results of the logistic regression model using the model's *predict_proba* method.
- Now rank two new data records using the model:
  - The following values are for each of these attributes: 'sepal length', 'sepal width', 'petal length', 'petal width'
  - [[5.8,2.8,5.1,2.4],[6.0,2.2,4.0,1.0]]