Oleksandr Romanko, Ph.D.
Associate Director, Financial Risk Quantitative Research, SS&C Algorithmics
Adjunct Professor, University of Toronto

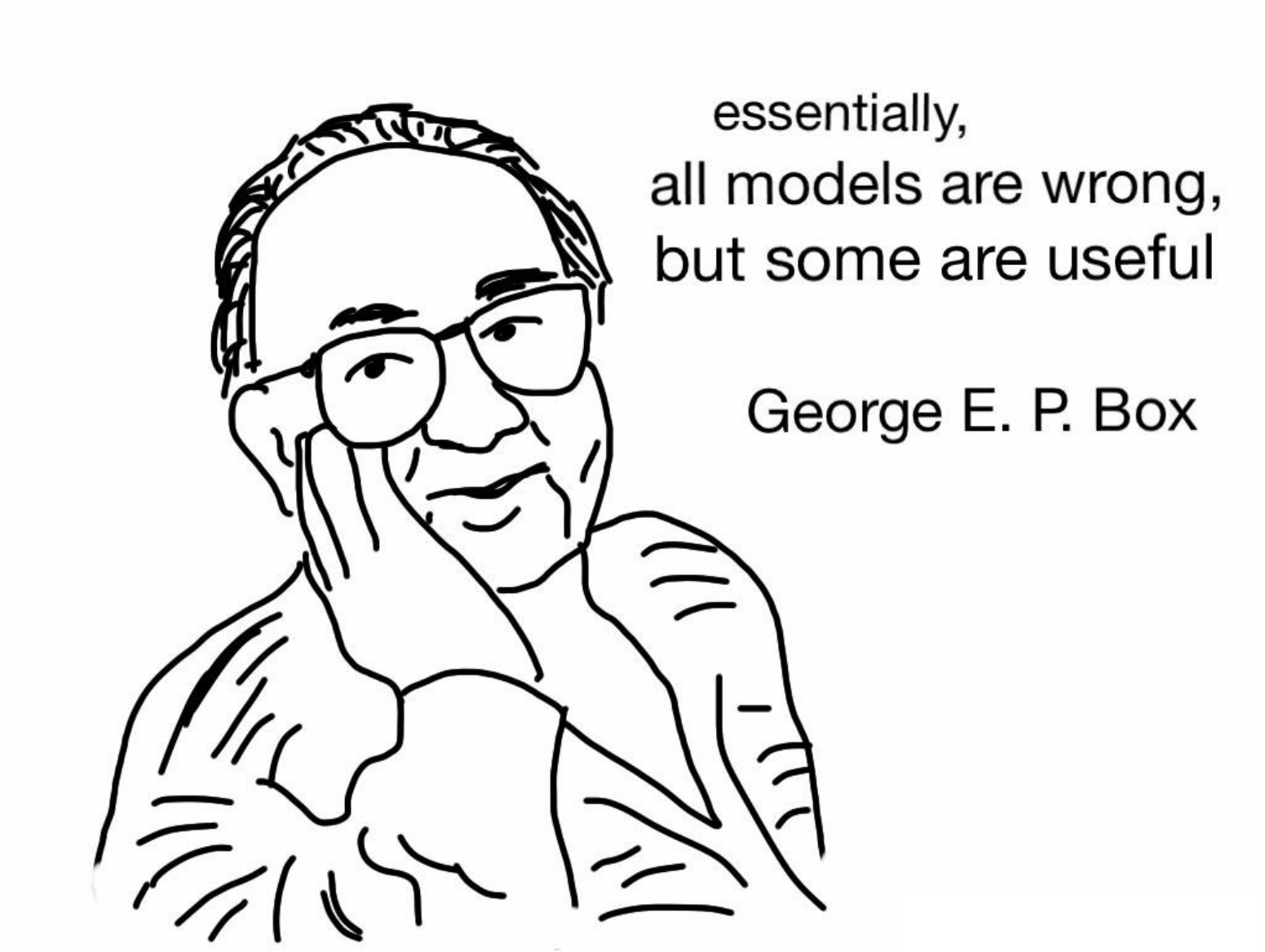
MIE1624H – Introduction to Data Science and Analytics

Lecture 5 – Modeling and Regressions

University of Toronto
October 10, 2023
October 17, 2023

Modeling

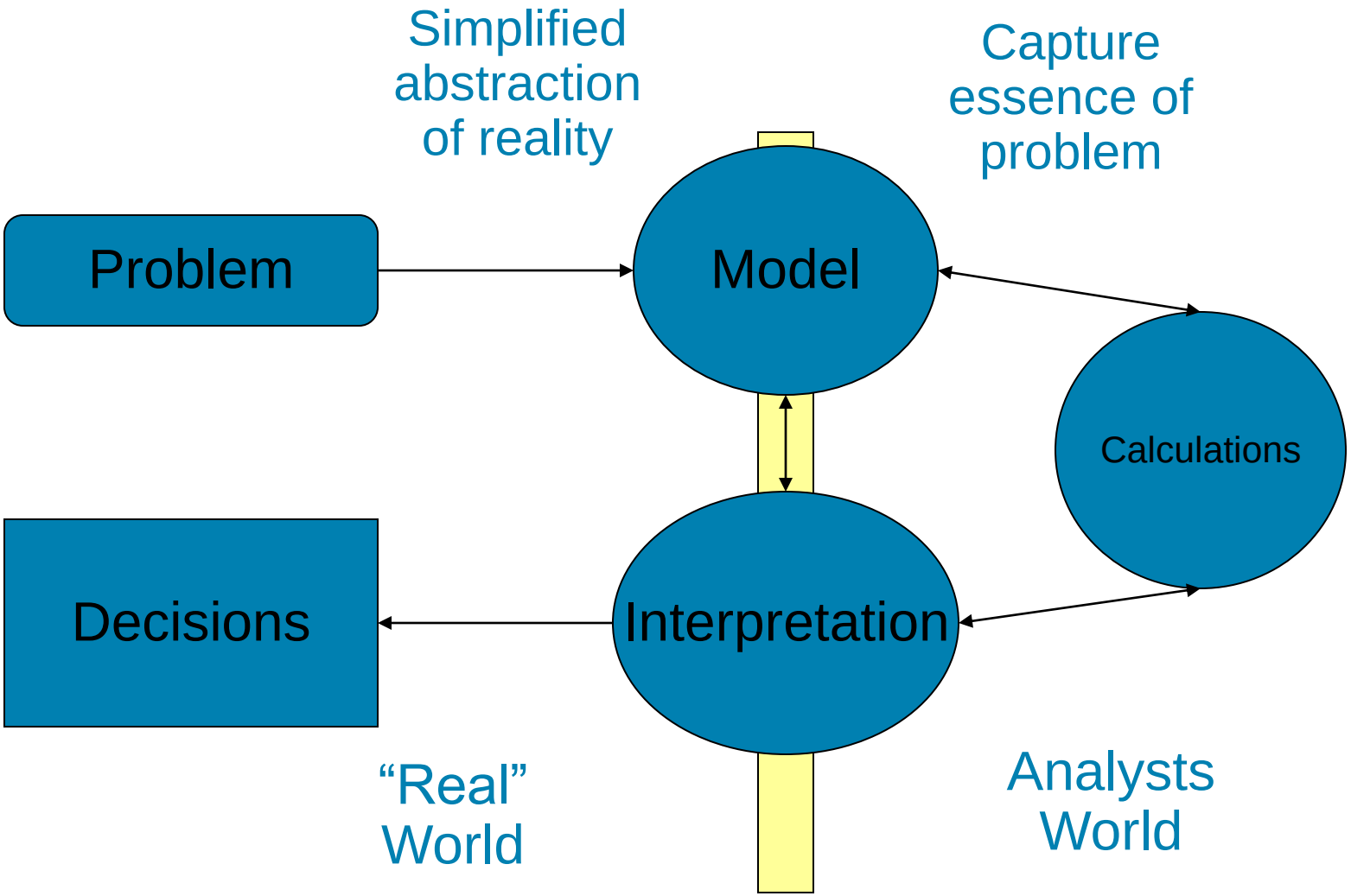
Models


essentially,
all models are wrong,
but some are useful
George E. P. Box

Models

- Simplified representation or abstraction of reality
- Capture essence of system without unnecessary details
- Models tailored for specific types of problems
- Models help us understand the world
  - Prediction (What if?)
  - Optimization (What's best?)
  - Clustering (How similar?)
- Often models much easier, faster, and cheaper to experiment with than the real system

Models and reality



From Monahan, G., "Management Decision Making", Cambridge University Press, 2000

Why do we model for decision making?

- **Building model forces detailed examination and thought about a problem**
  - structures our thinking
  - must articulate our assumptions, preconceived notions
  - model building may illuminate solution without actually using the model
- **Searching for general insights**
  - form of relationship between key variables involved in decision
  - importance of various parameters on decisions
- **Looking for specific numeric answers to a decision making problem**
  - If we add 1 lab tech between 7a-3p, how much reduction can we expect in test turnaround time?
- **Find the best way to do something**
  - Which routing schedule minimizes our delivery costs?

Types of models

- **Physical** – cars, buildings
- **Diagrams** – flow chart, decision trees, influence diagrams
- **Statistical** – regression equation, probability distribution
- **Mathematical** – queuing model, scheduling model
- **Computer simulation**
- **Computational** – neural networks, genetic algorithms

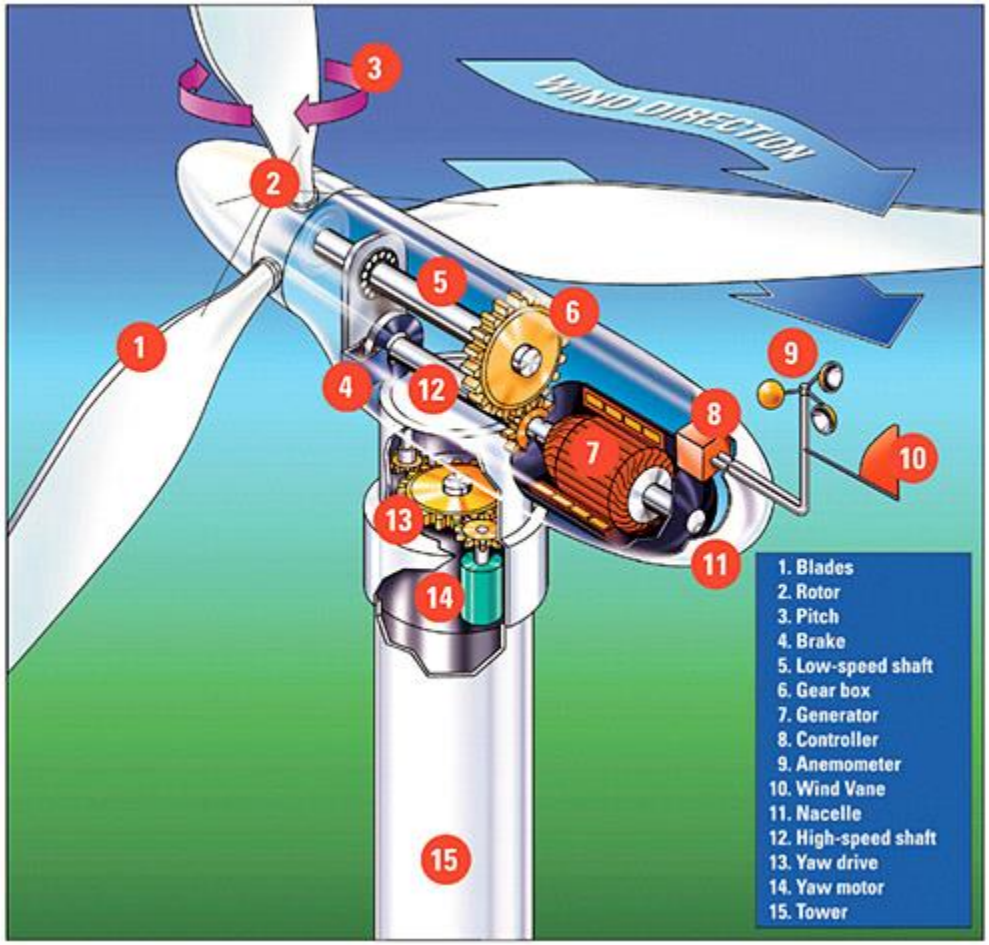
A 7 step (idealized) modeling process

- Define the problem
  - "exploring the mess"
- Observe system / collect data
- Formulate model(s)
  - much "art and craft"
- Verify/validate model and use for prediction and exploration of system being modeled
- Use model to help select among alternatives
- Present results to decision makers
- Implement solution and evaluate outcomes

📖 Predictive Maintenance

Predictive maintenance – before big data

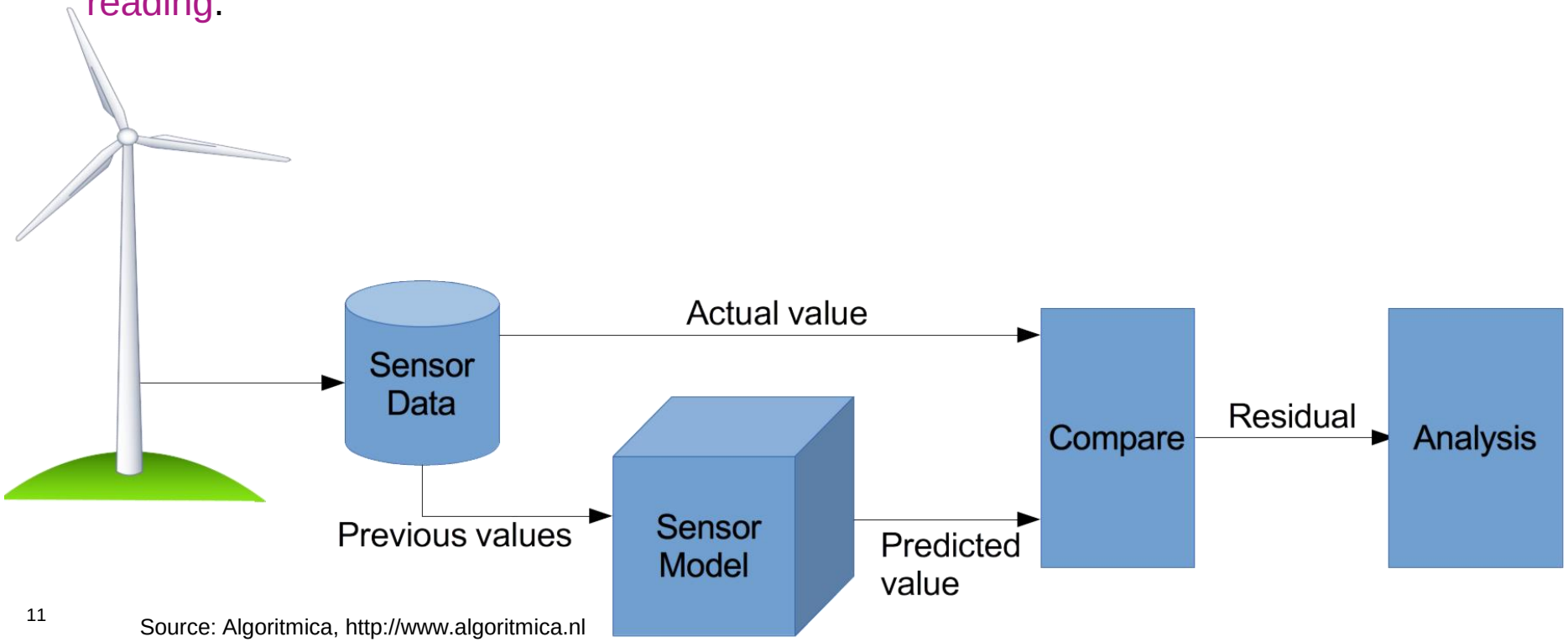
- Wind turbines are big and expensive machines, so keeping them running smoothly helps keeping their operational cost down. To do that, you need to be able to anticipate failures in heavy and expensive parts like the gearbox, generator and main shaft.
- **Physical models** explicitly describes the turbine design using detailed knowledge of its physical characteristics. A physical model has to be calibrated by an expert.
- Preventive maintenance saves money:
  - Shorter downtime and less lost production
  - Better planning of people and materials
  - Cheaper repairs



Source: Algoritmica, http://www.algoritmica.nl

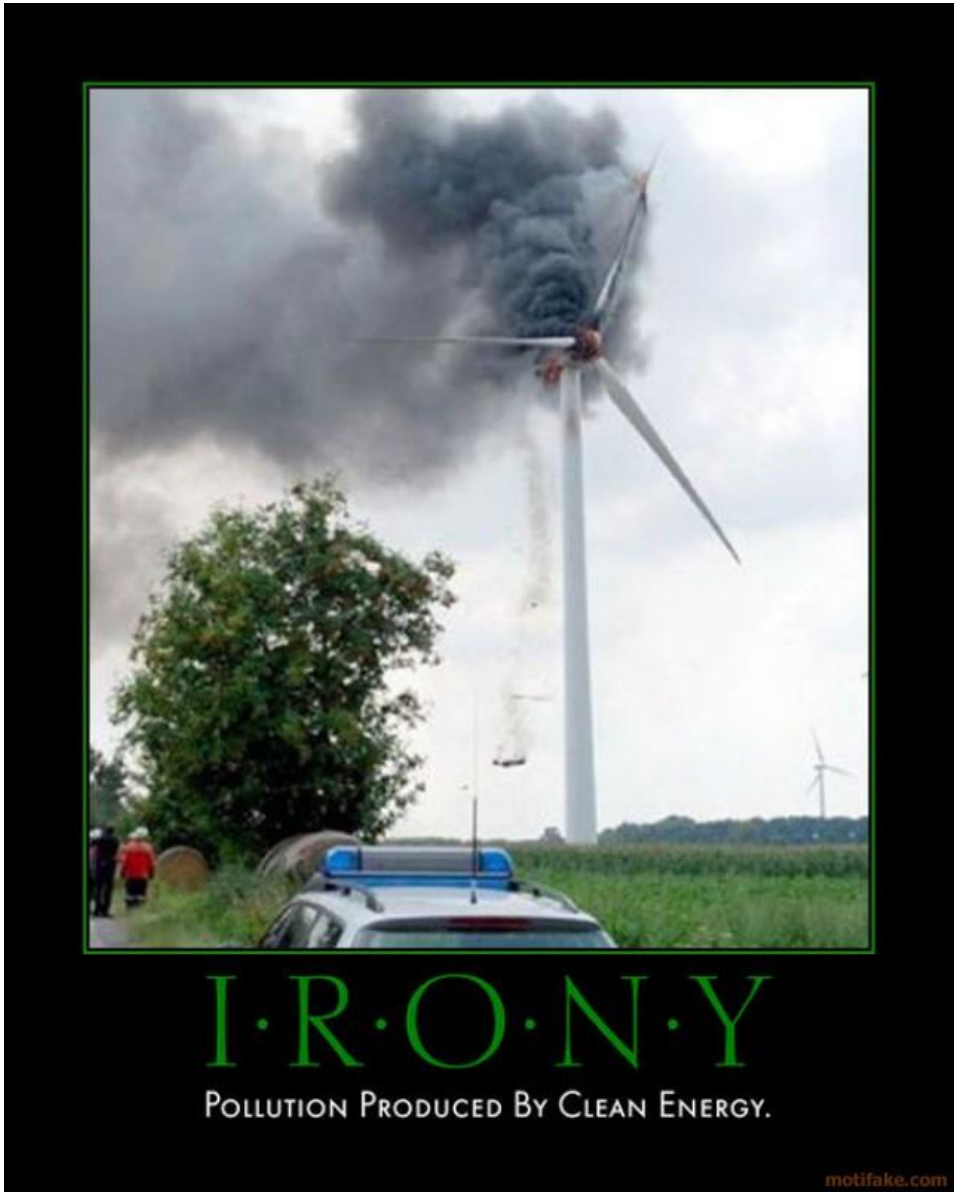
Predictive maintenance – big data era

- In the big data era wind turbines have an array of sensors that measure temperatures, pressures, voltages, currents, and blade angles.
- **Data-driven approach**: a model learns the relationship between the various sensor readings based on the training data. To create such a sensor model we apply machine learning, i.e. one or more algorithms that use a set of training examples to learn a predictive model. The model can be trained by a non-turbine expert. The model then calculates its predicted value and compares it with the actual sensor reading.



Source: Algoritmica, http://www.algoritmica.nl

Environmental risk management



📖 Model Examples

Pit stop analytics

F1 analytics based on past experience

Calculations showed that time spent changing tires and refilling the tank was more than offset by the improved performance of the car on the track.

1. Softer tires stuck to the track better during turns than their harder cousins, though they wore out more quickly.
2. Less gas in the tank translated into a lighter, and therefore faster, car.

Fitting room analytics

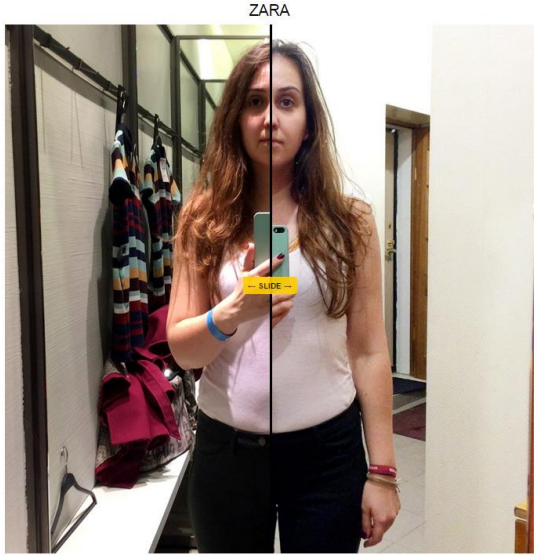
Good

SELA

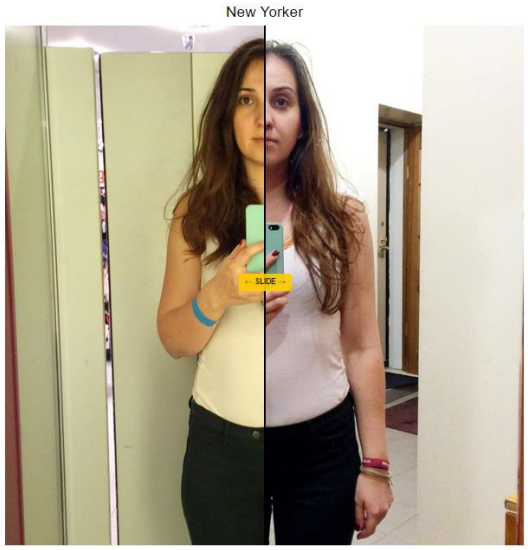
Mango

Bad

ZARA



New Yorker



Source: Adme

Shortest path or most beautiful path?

Forbes

Amit Chowdhry Contributor

I cover noteworthy technology, startups, and gadgets

Opinions expressed by Forbes Contributors are their own.

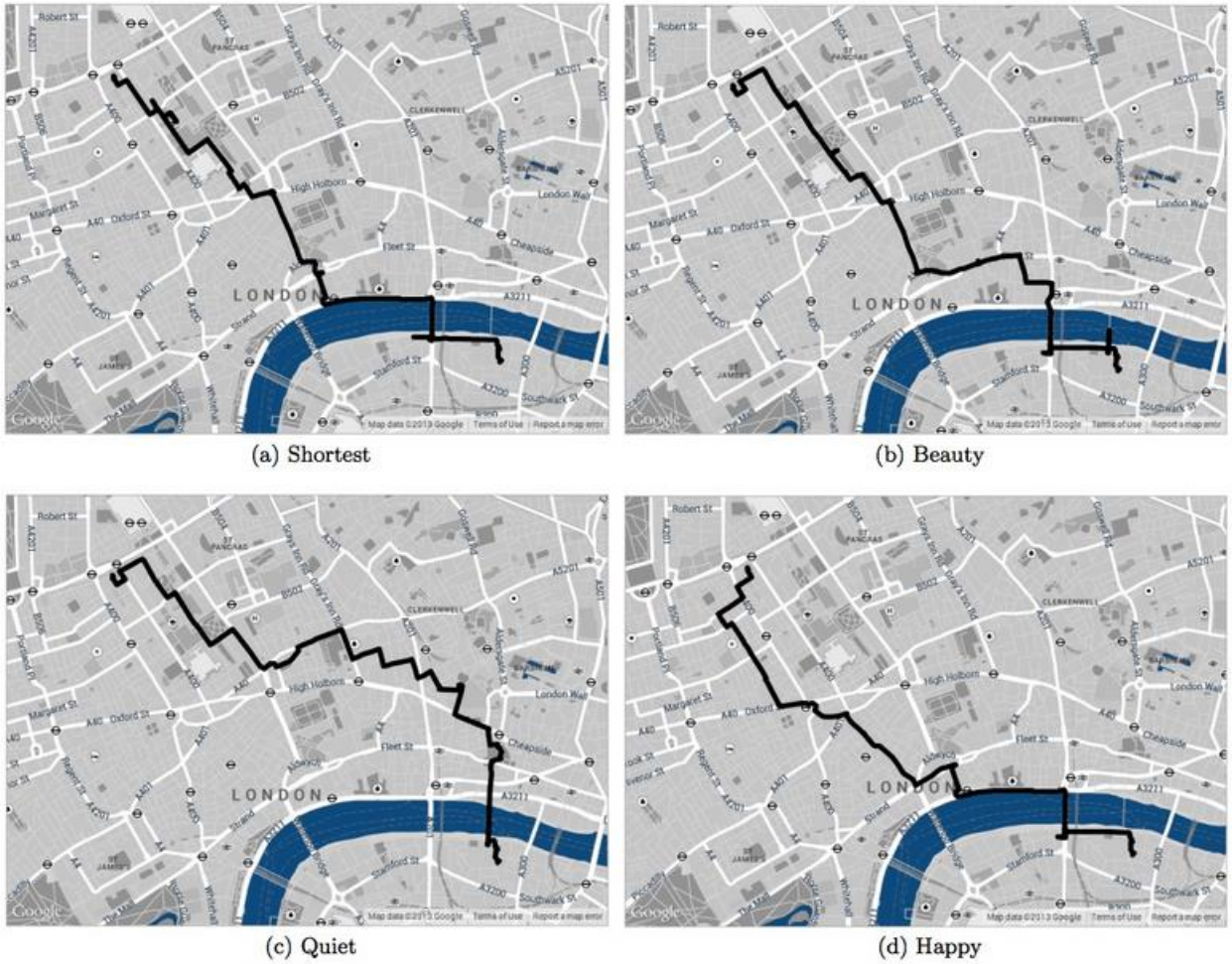
TECH 7/09/2014 @ 6:42PM | 1,273 views

Yahoo! Researchers Have Developed A GPS Algorithm To Find 'Emotionally Pleasant' Routes

+ Comment Now

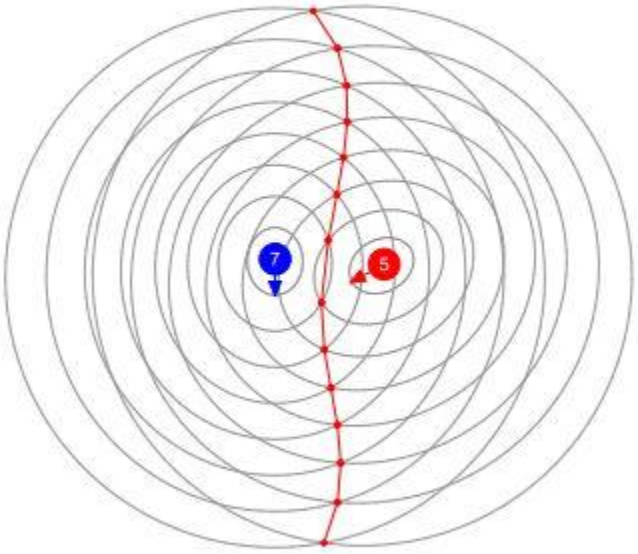
Mapping services and GPS devices are generally used for finding the shortest routes between two points while walking around or driving a car. Over the last few years, GPS technology has improved with features like the ability avoid tolls and construction. Now a team of researchers at Yahoo Labs in Barcelona, Spain want mapping services to offer pedestrians with a way to find the most scenic routes while heading towards their destination.

Shortest path or most beautiful path?

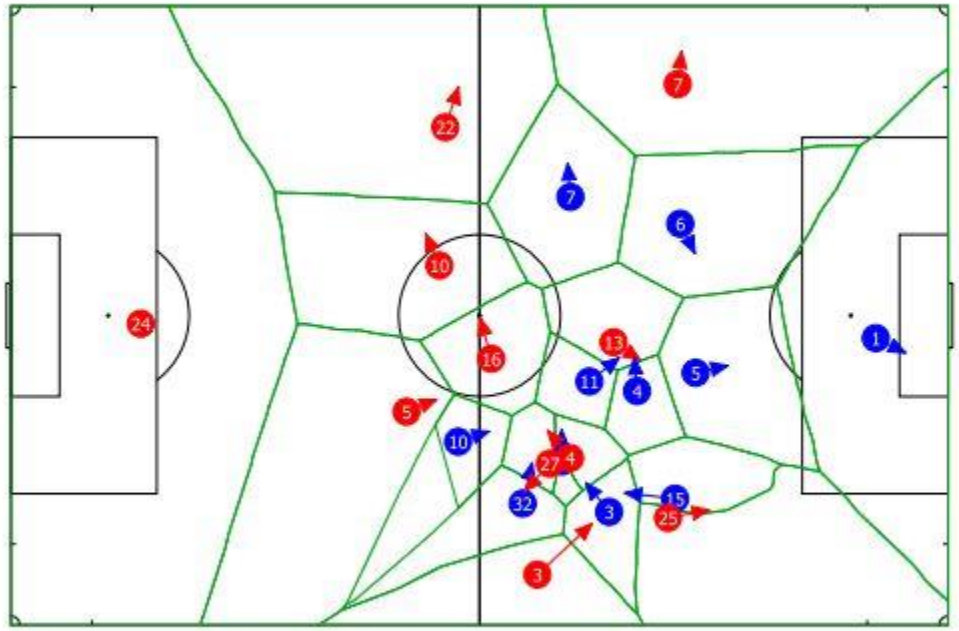


Routes plotted in London based on different algorithms. Credit: Yahoo!

Sports analytics

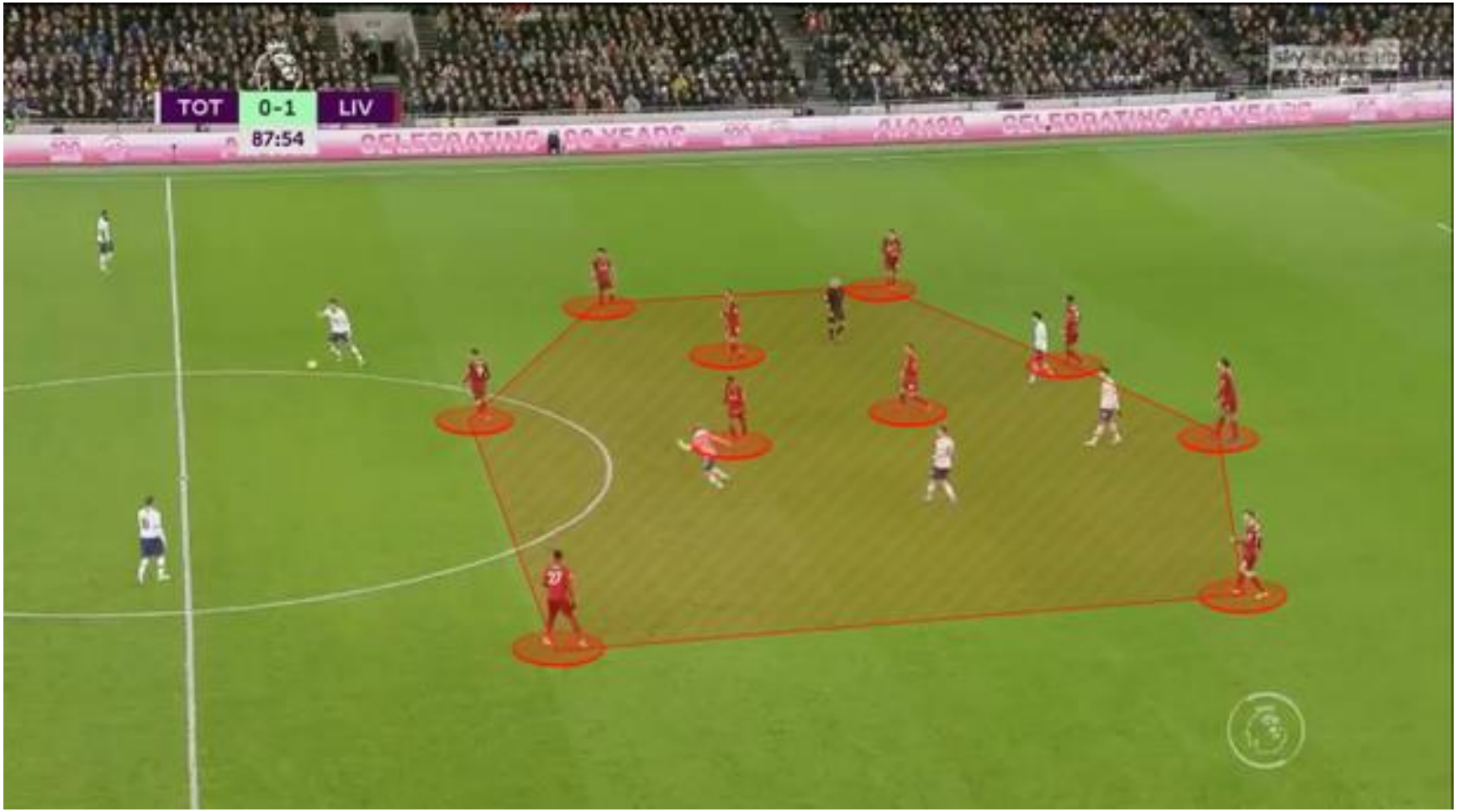


(a)



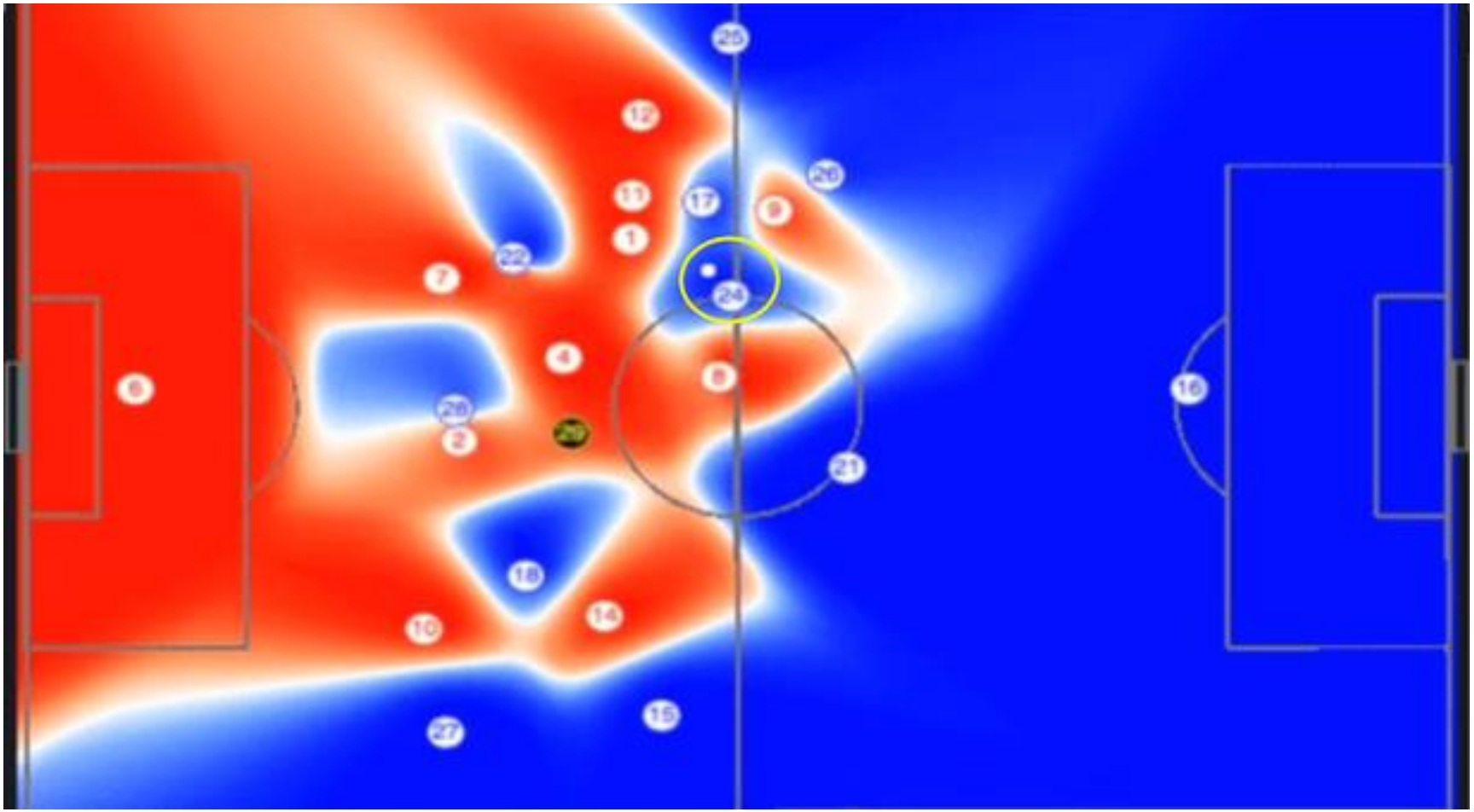
(b)

Sports analytics



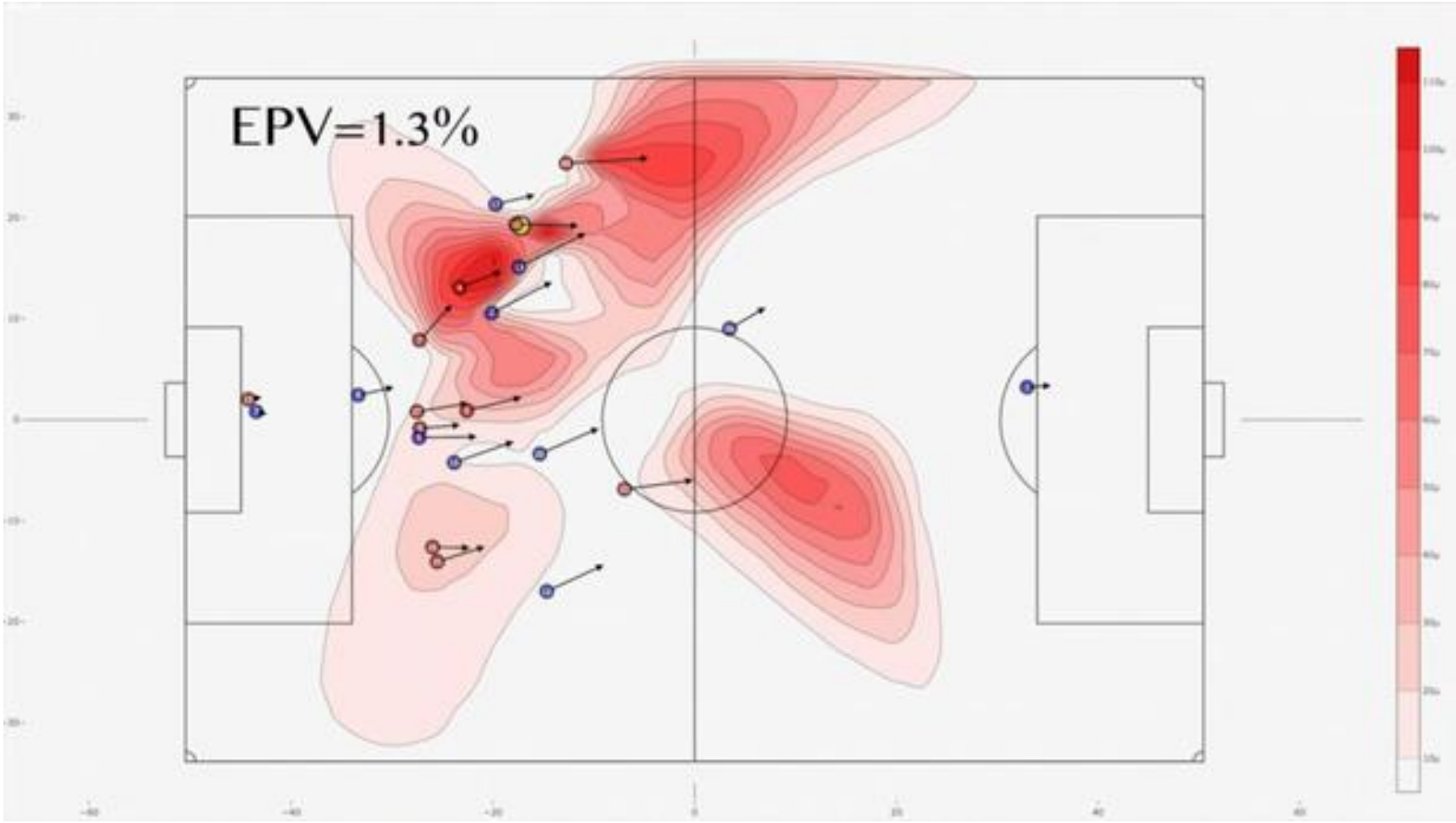
Source: https://www.liverpool.com/liverpool-fc-news/features/liverpool-transfer-news-jurgen-klopp-17569689

Sports analytics



Source: https://www.liverpool.com/liverpool-fc-news/features/liverpool-transfer-news-jurgen-klopp-17569689

Sports analytics



Source: https://www.liverpool.com/liverpool-fc-news/features/liverpool-transfer-news-jurgen-klopp-17569689

Personalized health analytics – 23andMe

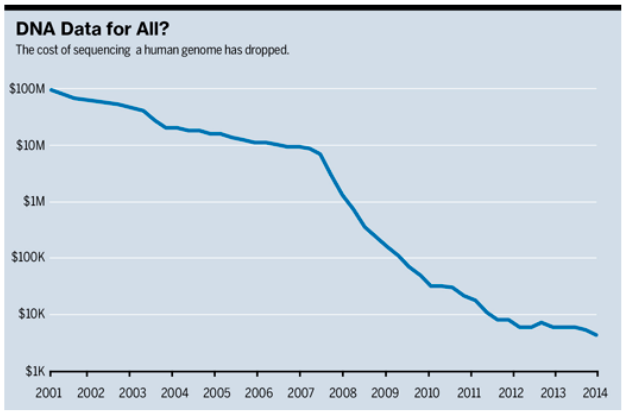
Personalized health analytics – 23andMe

BUSINESS REPORT Data-Driven Health Care

23andMe Tries to Woo the FDA

The DNA testing firm hopes a more coöperative approach with regulators will get its business back on track.

By Robert D. Hof on July 21, 2014



MIT Technology Review

Anne Wojcicki bounds into a conference room in Mountain View, California, straight from a five-mile ride from home on an elliptical bike. The 40-year-old cofounder and CEO of the consumer genetic testing firm 23andMe is breathless, and not just because of the workout. On this warm day in mid-June, Wojcicki is "super-excited" about an announcement scheduled for two days hence: the Food and Drug Administration has agreed to review a health-related genetic report the company wants to make available to customers.

It's the first step out of the FDA's doghouse for 23andMe. For $99, the company analyzes key components of a person's DNA from a vial of saliva, but last November the federal agency issued a testy warning letter barring it from marketing its service. The FDA said that by selling consumers a test and health reports that outlined their chances of getting dozens of diseases, plus their likely response to various drugs, 23andMe was effectively selling a medical device. That requires explicit approval – and the FDA said 23andMe hadn't come close to providing enough evidence that its test provides accurate, reliable health assessments.


MIT Technology Review
BUSINESS REPORT

Data-Driven Health Care

New technologies promise a flood of molecular, environmental, and behavioral patient information. Will all that data make medicine better?

CONTENTS
The Big Question
More Phones, Fewer Doctors
IBM Aims to Make Medical Expertise a Commodity
23andMe Tries to Woo the FDA
Mobile Health Monitoring Devices
Mobile Health's Growing Pains
Plus: Citi's Crash, Data in Action at Mayo, Pharma's new transparency, and more

The Big Question
Can Technology Fix Medicine?

Medical data is a hot spot for venture investing and product innovation. The goal: better care.

● After decades as a technological laggard, medicine has entered its data age. Mobile technologies, sensors, genome sequencing, and advances in analytic software now make it possible to capture vast amounts of information about our individual makeup and the environment around us. The sum of this information could transform medicine, turning a field aimed at treating the average patient into one that's customized to each person and shifting more control and responsibility from doctors to patients.

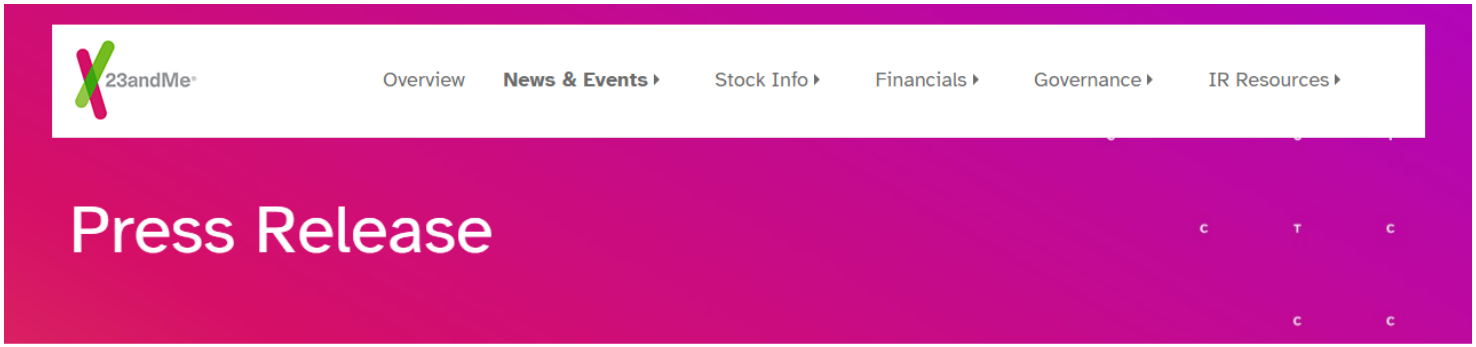
The question is: can big data make health care better?

"There is a lot of data being gathered. That's not enough," says Ed Martin, interim director of the Information Services Unit at the University of California, San Francisco, School of Medicine. "It's really about coming up with applications that make data actionable." →

1

CONTENTS

Personalized health analytics – 23andMe



23andMe Receives FDA Clearance for Direct-to-Consumer Genetic Test on a Hereditary Prostate Cancer Marker

January 10, 2022

PDF Version

The clearance allows 23andMe to report on the G84E mutation in the HOXB13 gene, clinically shown to significantly increase the risk of developing prostate cancer in men with the mutation

SUNNYVALE, Calif., Jan. 10, 2022 (GLOBE NEWSWIRE) -- 23andMe Holding Co. (Nasdaq: ME) ("23andMe"), a leading consumer genetics and research company, today received FDA clearance for a genetic health risk report on a hereditary prostate cancer marker.

This is the Company's third cancer risk report clearance, following the FDA's prior authorization for 23andMe's BRCA1/BRCA2 (Selected Variants) Genetic Health Risk report and its clearance for MUTYH-Associated Polyposis (MAP), a hereditary colorectal cancer syndrome. These two reports along with the new Hereditary Prostate Cancer (HOXB13-Related) report have been included by the FDA in a single "Cancer Predisposition Risk Assessment System" regulation.

These three 23andMe reports are the only direct-to-consumer genetic health risk reports for inherited cancers that have been authorized by the

Personalized health analytics – 23andMe

THE RISE, FALL AND RISE AGAIN OF 23ANDME

HOW ANNE WOJCICKI TOOK THE START-UP FIRM FROM THE BRINK OF FAILURE TO SCIENTIFIC PRE-EMINENCE.

BY ERIKA CHECK HAYDEN

174 | NATURE | VOL 550 | 12 OCTOBER 2017

Personalized health analytics – 23andMe

23andMe genetics just got personal.

Search 23andMe Go | Log in | Register Your Kit | Blog | Help

welcome | ancestry | health | how it works | store

Start filling in the gaps with your DNA

"Because I had given my doctor information from 23andme, he got to a diagnosis much faster. 23andme saved my life." Kirk C.

$99* Our new low price for all! Was $199

Order Now »

*Requires a 1-year commitment to the Personal Genome Service® at $9/mo. **Order for $399** without commitment.

❶ Get Your Kit

❷ Provide Saliva

❸ Learn About Yourself

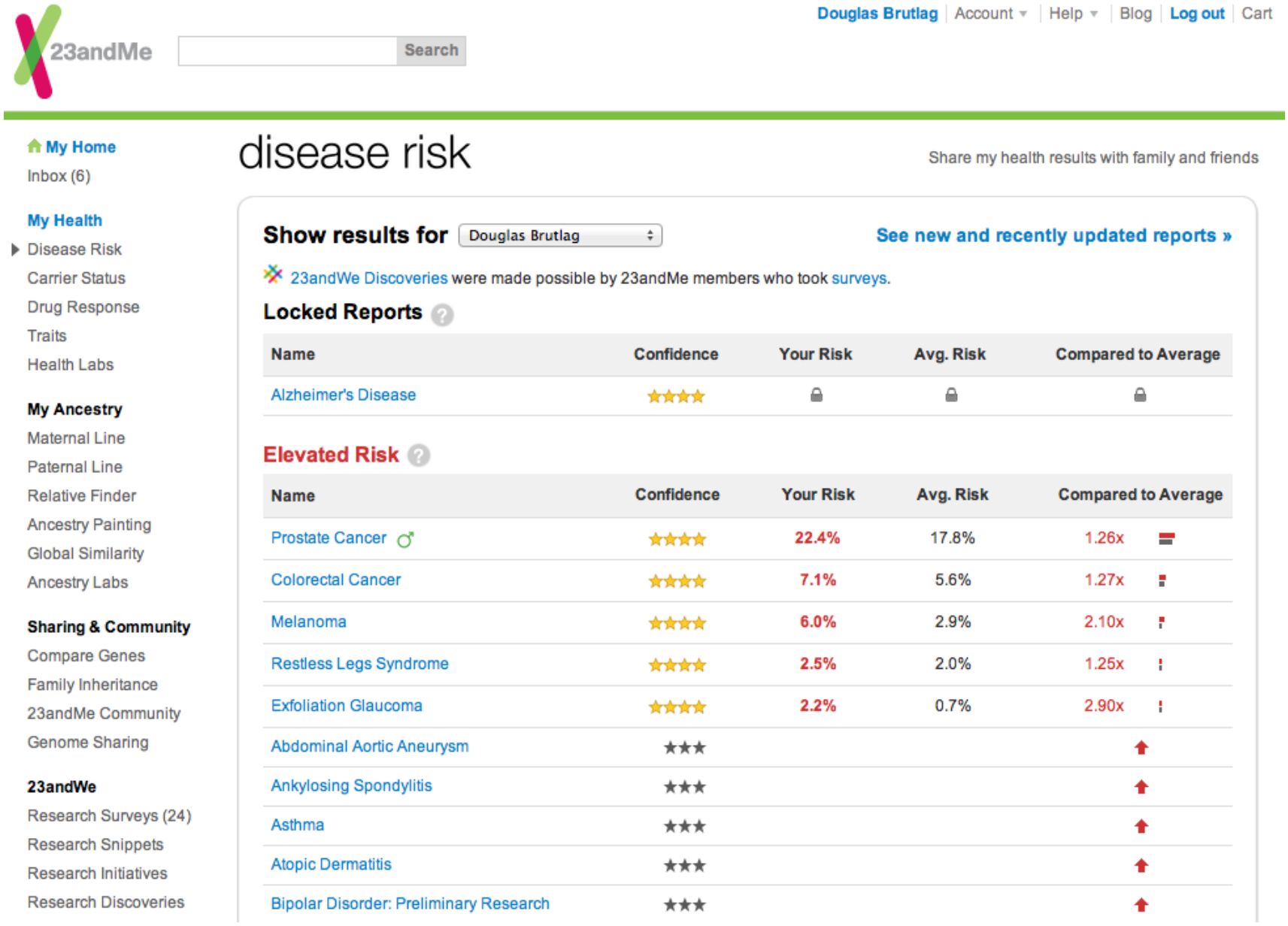
❹ Get Monthly DNA Discoveries

Gain insight into your traits, from baldness to muscle performance. Discover risk factors for 97 diseases. Know your predicted response to drugs, from blood thinners to coffee. And uncover your ancestral origins. start tour »

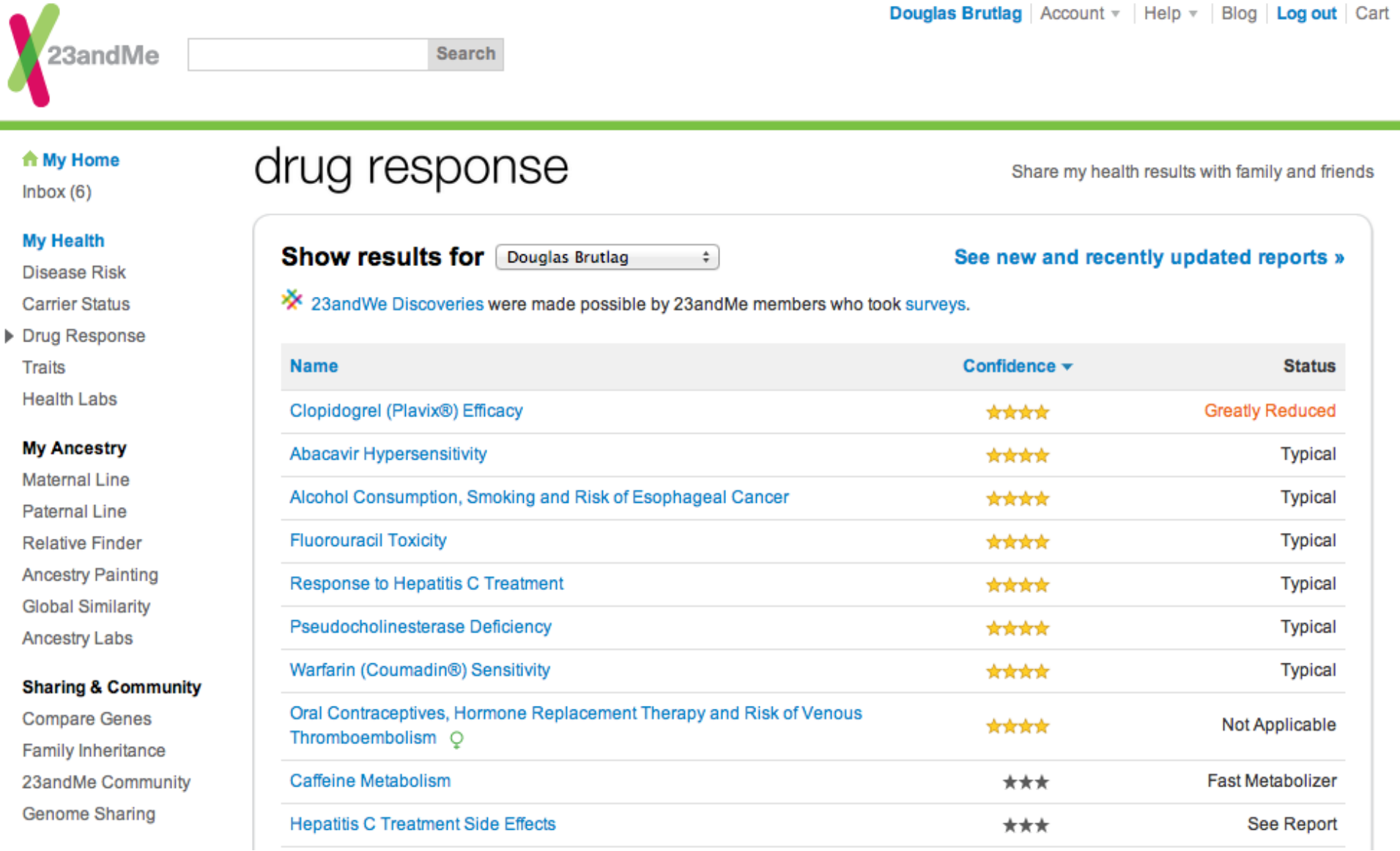
Overview | Discover Health & Ancestry | Keep Your Doctor Informed | Participate In Research

Personalized health analytics – 23andMe

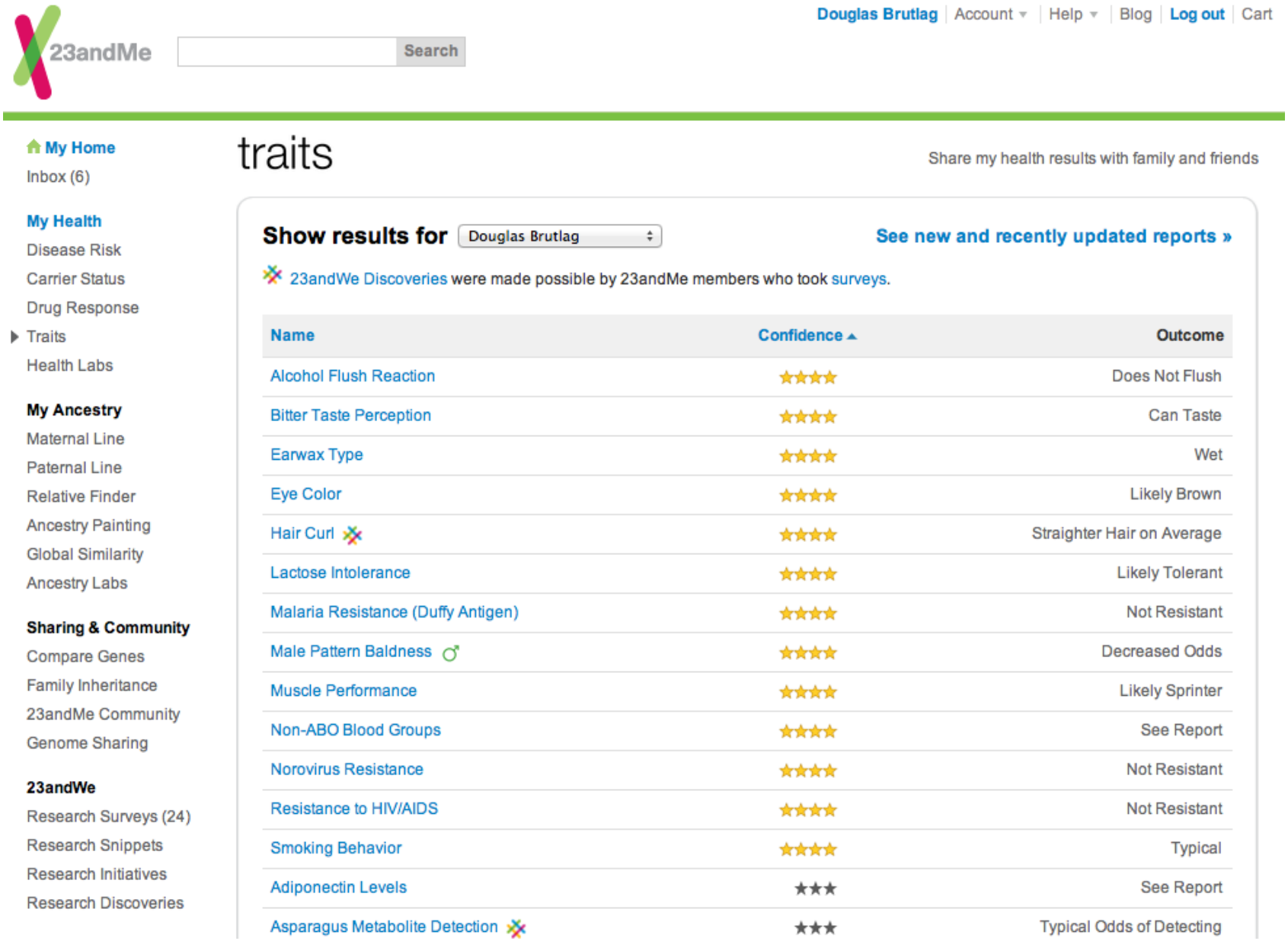
Personalized health analytics – 23andMe



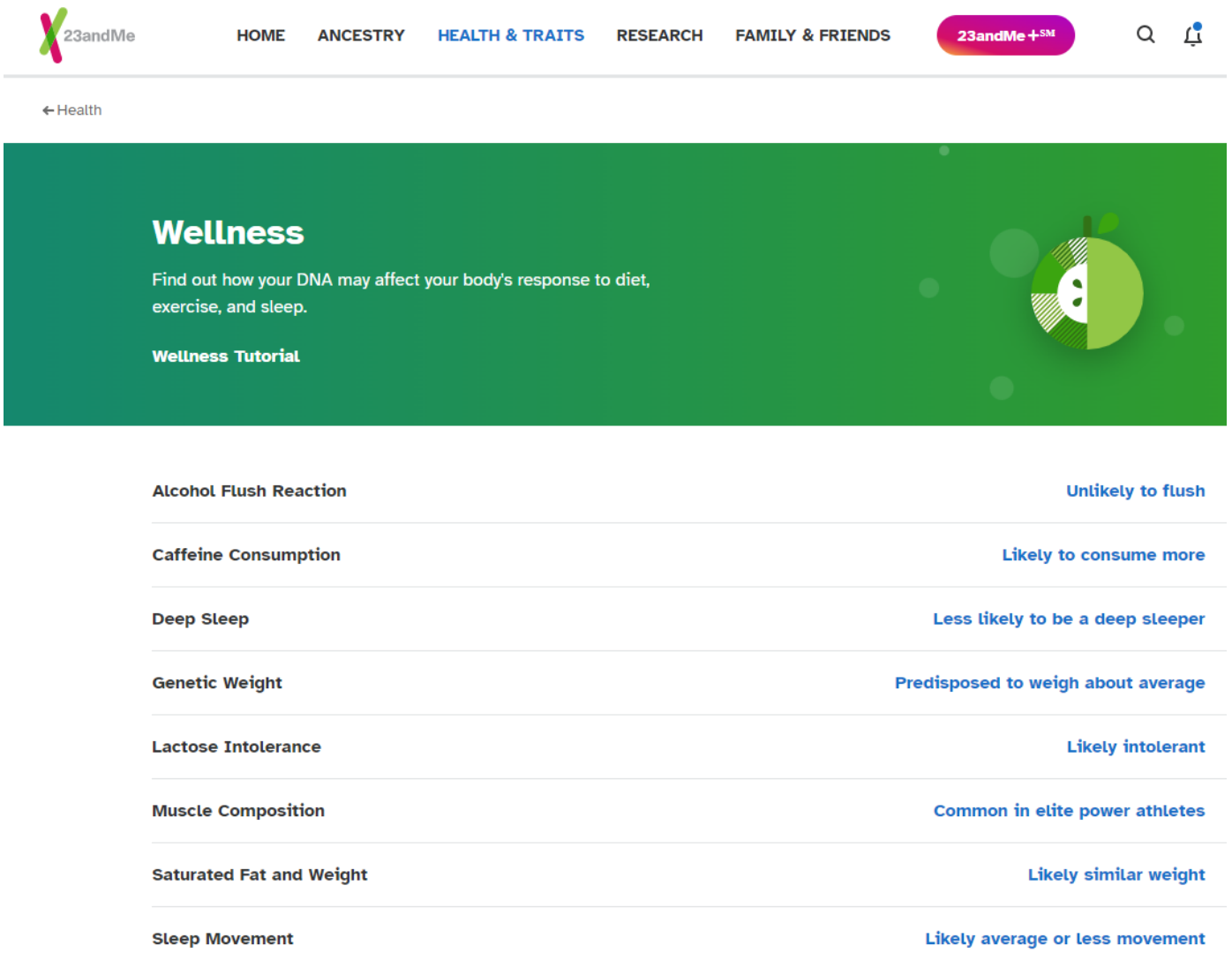
Personalized health analytics – 23andMe



Personalized health analytics – 23andMe



Personalized health analytics – 23andMe



Personalized health analytics – 23andMe

23andMe genetics just got personal.

brutlag | genetics 101 | blog | help | sign out

me
- My Gene Journal (60)
- Browse Raw Data

family & friends
- Compare Genes
- Family Inheritance

my ancestors
- Maternal Line
- Paternal Line
- Ancestry Painting
- Global Similarity

account
- My Profiles
- Genome Sharing
- Settings
- Help/Contact Us

maternal line

Your mitochondrial DNA determines your maternal haplogroup. What is a haplogroup?

Map | History | Haplogroup Tree

Maternal Haplogroup: U5

Locations of haplogroup U5 circa 500 years ago, before the era of intercontinental travel.

Haplogroup U5 arose among early colonizers of Europe around 40,000 years ago; maternal descendants of those early colonizers persist in the region to this day. After the last Ice Age two subgroups of U5 expanded across Europe and into northern Africa and the Near East. Today, one subgroup, U5b1b, is shared by groups as diverse as the northern African desert-dwelling Berbers and the Scandinavian Arctic-dwelling Saami, also known as the Lapps.

Haplogroup: U5, a subgroup of U

Age: 40,000 years

Region: Europe, Near East, North Africa

Populations: Basques, Saami (Lapps) of northern Scandinavia

Highlight: Though primarily a European haplogroup, U5 was recently found in mitochondrial DNA extracted from the remains of a 6th-century AD Chinese chieftain.

Your Family and Friends

K1a1b1a	Simone Brutlag
U5b2	Douglas Brutlag
L3e	Nigerian Man
D5a2	Chinese Man
D4e2	Japanese Man

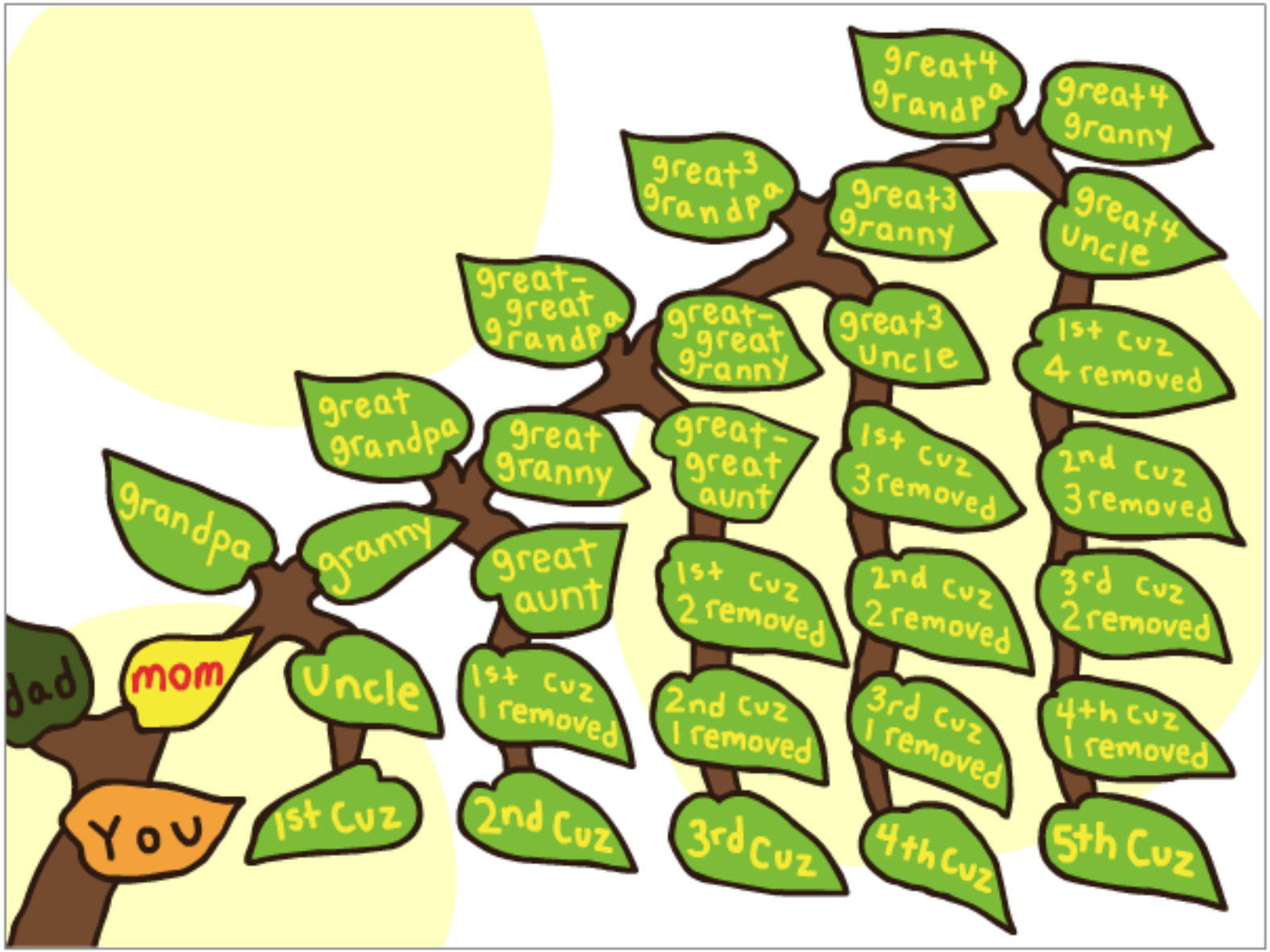
Personalized health analytics – 23andMe

relative finder

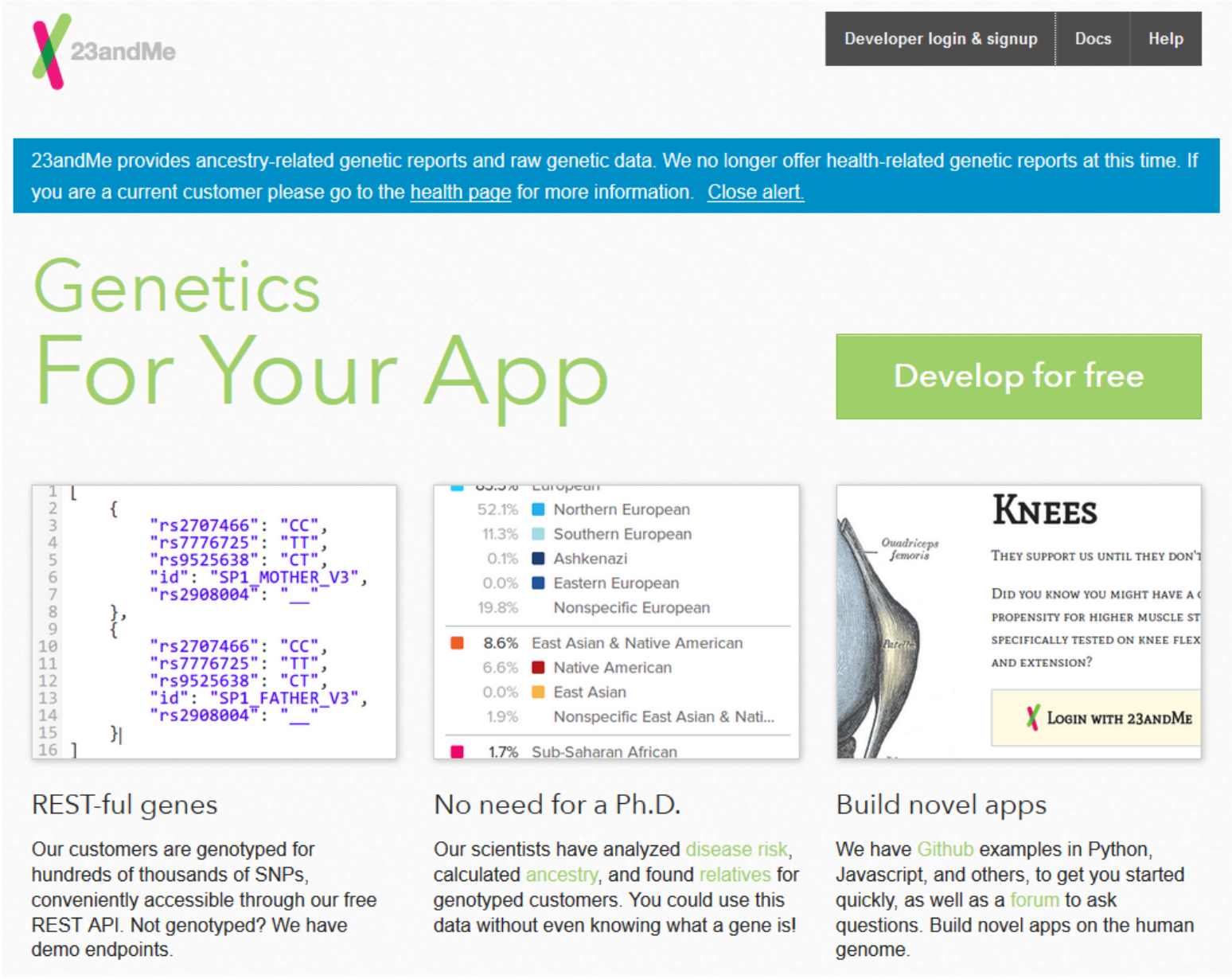
search matches | sort by relationship | 25 per page | « ← 1 – 25 of 309 matches → »

Male	You	U5b2a, E1b1b1a2*	Update Your Profile
Benjamin Brutlag Male, b. 1980	Son 47.7% shared, 22 segments	United States, Southern Europe, K1a1b1a, E1b1b1a2*	Sharing Genomes Send a Message
Pauline Brutlag Female	Daughter 53.1% shared, 25 segments	United States, Northern Europe, K1a1b1a	Sharing Genomes Send a Message
Male	3rd to 5th Cousin 0.47% shared, 3 segments	H3, R1a1a*	Send an Introduction
Larry Vongroven Male	3rd to 5th Cousin 0.54% shared, 2 segments	United States, Alen, Norway, Haltalen, Norway, Voss, Norway, 8 more, Northern Europe, Vongroven (Vongraven), Bakken, Goodno, 10 more, U4b1a2, R1a1a	Introduction Received Respond
Female	3rd to 5th Cousin 0.47% shared, 2 segments	K1a10	Send an Introduction
Male	3rd to 5th Cousin 0.34% shared, 2 segments	I2a, R1a1a*	Send an Introduction
Male	3rd to 6th Cousin 0.39% shared, 1 segment	U5b1b1a, R1b1b2a1a1	Send an Introduction
Male	3rd to 6th Cousin 0.36% shared, 1 segment	H2a5, R1b1b2a1a1*	Send an Introduction
Male	3rd to 6th Cousin 0.31% shared, 2 segments	H7a, D1*	Send an Introduction
ivan otterness Male, b. 1938	3rd to 6th Cousin 0.30% shared, 2 segments	United States, Minnesota, Wisconsin, Norway, Northern Europe, Otterness, Brandsness, Flekke, 4 more, K1a10, I2b1	Send a Message
Female	3rd to 6th Cousin 0.29% shared, 2 segments	U4b1a1	Introduction Received Respond

Personalized health analytics – 23andMe

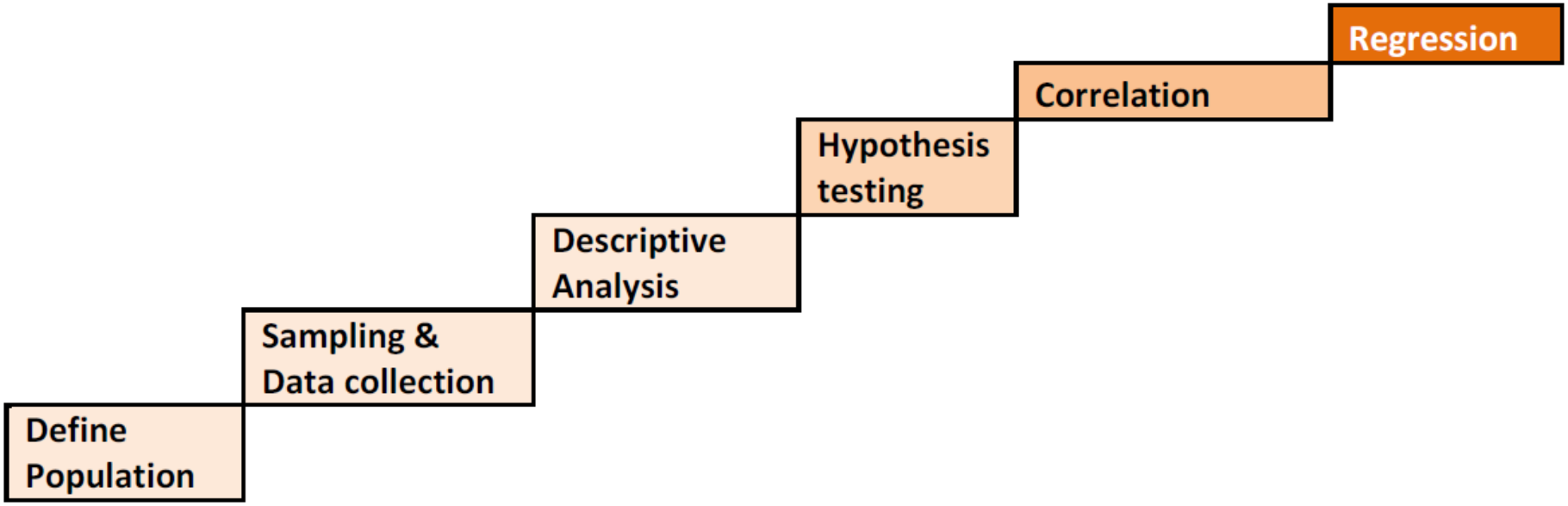


Personalized health analytics – 23andMe



🕮 Linear Regression – Statistical Perspective

Linear regression – statistical perspective



Source: Murtaza Haider "Getting Started with Data Science: Making Sense of Data with Analytics"

Linear regression – statistical perspective

How is cab fare calculated?

You sit in the cab and see a base fare that is constant	• $3.25
The fare increases with each additional 143 m at a certain rate	• $0.25 per 143 m
The fare increases for each 29 seconds of taxi being idle	• $0.25 per additional 29 seconds
For each passenger in excess of 4	• $2.00

Source: Murtaza Haider "Getting Started with Data Science: Making Sense of Data with Analytics"

Linear regression – statistical perspective

Put it in a formula

- Fare = Constant + Dist_Rate x Distance + Wait_Rate x Time + FourPlus x Four or more Passengers

- Fare = 3.25 + 0.25 x Distance + 0.25 x Idle Time + 2 x Four_plus

- A sample trip
  - 3 people
  - 6 kms
  - 4 minutes
- Adjusting time and distance
  - 6000 m / 143 = 41.9 segments of 143 m
  - 4(60)/29 = 8.3 segments of 29 seconds

- **Fare = $3.25 + $0.25 x 41.9 + $0.25 x 8.3 + $2 x 0 = $15.8**

Source: Murtaza Haider "Getting Started with Data Science: Making Sense of Data with Analytics"

Linear regression – statistical perspective

We can do this by hand

Fare	Distance (km)	Time (seconds)	4+ pax
	$0.25 for 143 m	$0.25 for 29 seconds	$2 for 4+ pax
$ 15.8	6	240	0
$ 16.9	6.7	225	0
$ 23.1	8.5	350	1
$ 11.4	3.8	180	0
$ 12.1	3.25	135	1
$ 13.5	4.5	275	0

Source: Murtaza Haider "Getting Started with Data Science: Making Sense of Data with Analytics"

Linear regression – statistical perspective

Why to regress?

What if we don't know the rates?

Fare	~~$0.25 for 143 m~~ Distance (km)	~~$0.25 for 29 seconds~~ Time (seconds)	~~$2 for 4+ pax~~ 4+ pax
$ 15.8	6	240	0
$ 16.9	6.7	225	0
$ 23.1	8.5	350	1
$ 11.4	3.8	180	0
$ 12.1	3.25	135	1
$ 13.5	4.5	275	0
y	x_1	x_2	x_3

Regression estimates the unknown rates

Fare = Constant + Dist_Rate x Distance + Wait_Rate x Time + FourPlus x Four or more Passengers

$$y = \beta_0 + \beta_1 * distance + \beta_2 * time + \beta_3 * pax + \epsilon$$

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \beta_3 \cdot x_3 + \epsilon$$

Source: Murtaza Haider "Getting Started with Data Science: Making Sense of Data with Analytics"

Linear regression – statistical perspective

Things to remember

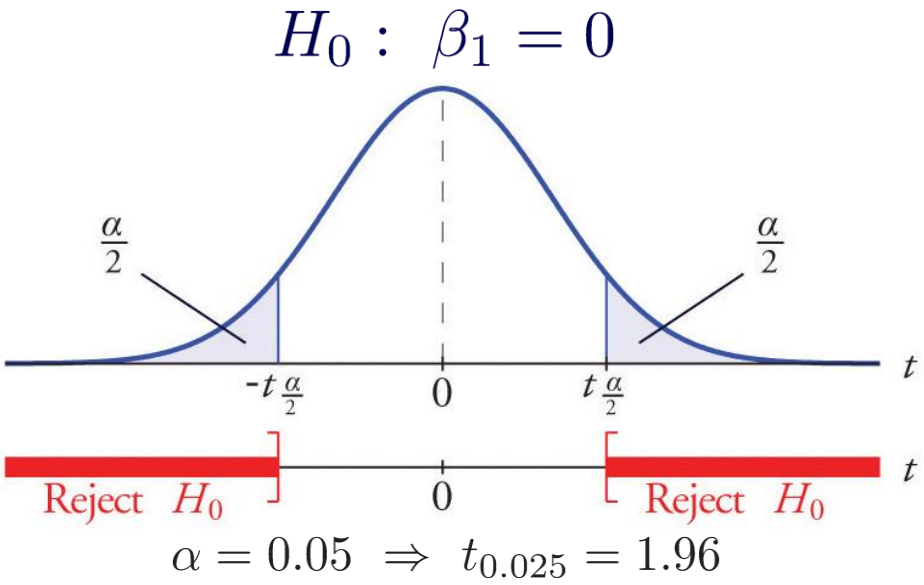
- **`R-squared`** (r^2): The overall fit of the model is determined by `R-squared` or the adjusted `R-squared`. Its value varies between 0 and 1. When comparing models, a higher `R-squared` means a better fit.
- **`P-value`**: The `p-value` is the probability associated with a statistical test. Usually, we use 95% level as our benchmark. Hence, we conclude a statistically significant finding when the `p-value` associated with a test is less than 0.05.
- **`F-test`**: When the `p-value` associated with the `F-test` is less than 0.05, we conclude that taken together the explanatory variables in the regression model are collectively different from 0.
- **`T-statistic`**: This evaluates the significance of an individual coefficient corresponding to a variable. If the `t-test` for a variable is greater than the critical value from the t-distribution (usually 1.96 for a two-tailed test; see Chapter 6 for details), we conclude that there exists a statistically significant relationship between the dependent and explanatory variable. Also, we can rely on the `p-value`, also reported in the regression output, for the corresponding `t-test`. If the `p-value` is less than 0.05, we can conclude a statistically significant relationship between the dependent and explanatory variables.

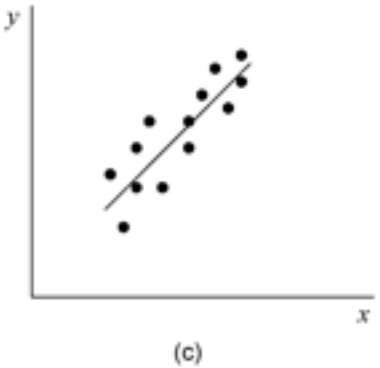
Source: Murtaza Haider "Getting Started with Data Science: Making Sense of Data with Analytics"

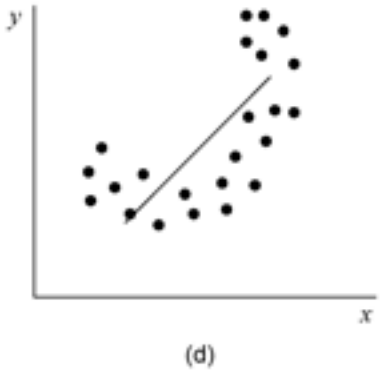
Linear regression – statistical perspective

Hypothesis testing

Null hypothesis: no linear relationship exists between independent variable ***x*** and dependant variable ***y***







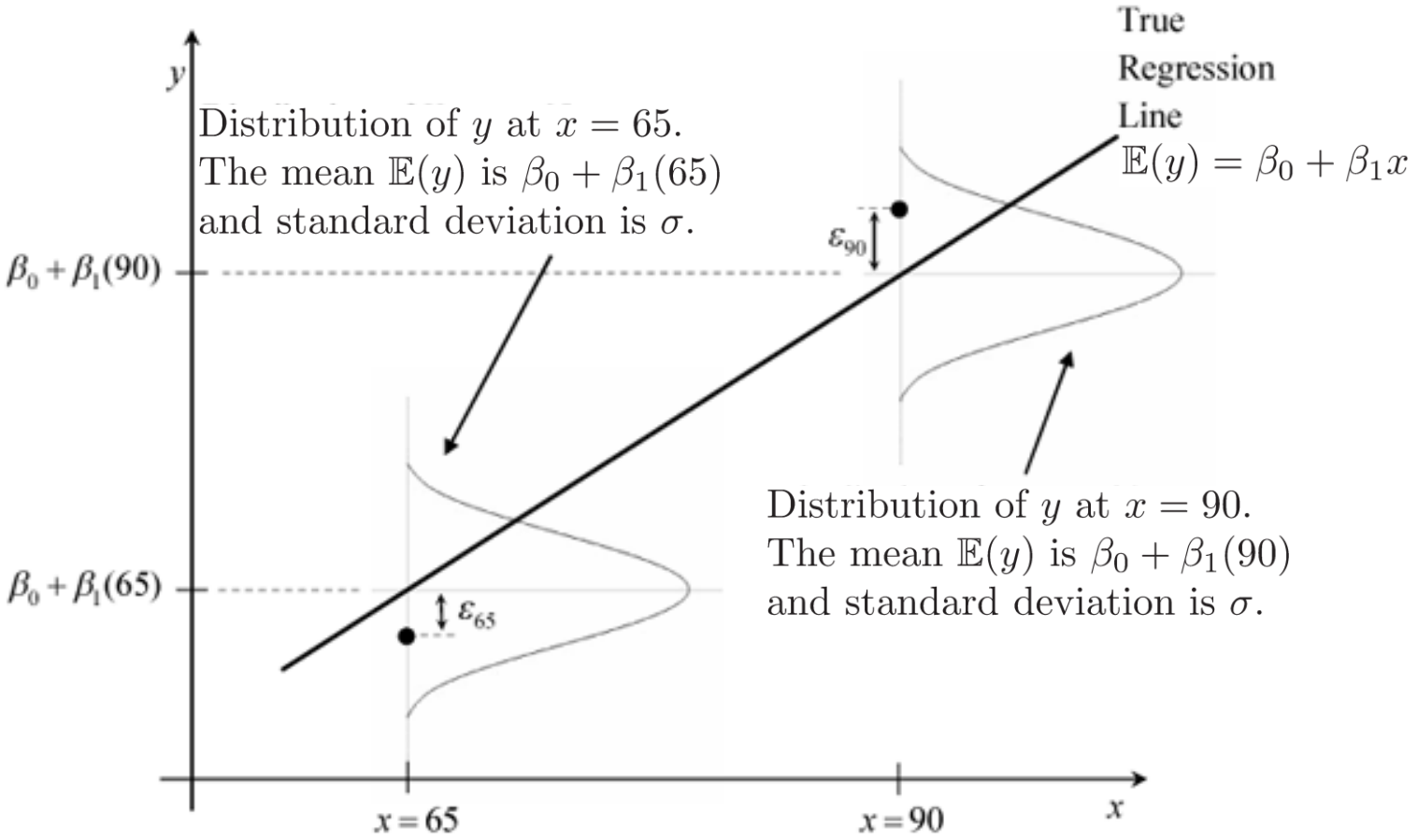
a) no linear relationship exists (H_0)
b) true relationship is not linear
c) linear relationship exists (H_0 rejected)
d) a higher order model may be needed

Linear regression – statistical perspective

Distribution of errors

Random error term ϵ is assumed to follow Normal distribution $\mathcal{N}(0, \sigma^2)$

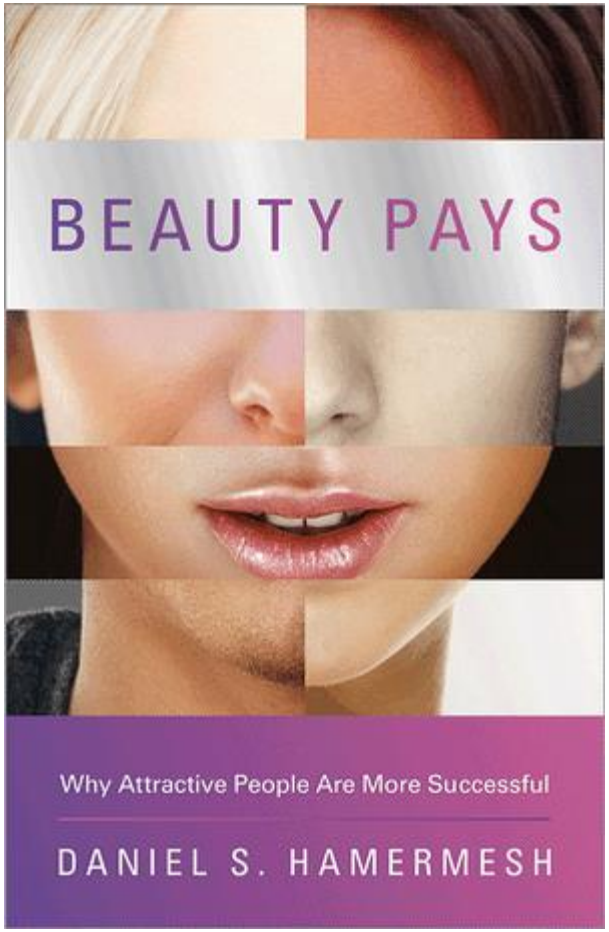
At any $x^{(j)}$, y follows Normal distribution with mean $\beta_0 + \beta_1 x^{(j)}$ and st.dev σ



Linear regression – statistical perspective

Applications

Do good looking professors get higher teaching evaluations?



NBER WORKING PAPER SERIES

BEAUTY IN THE CLASSROOM:
PROFESSORS' PULCHRITUDE AND
PUTATIVE PEDAGOGICAL PRODUCTIVITY

Daniel S. Hamermesh
Amy M. Parker

Working Paper 9853
http://www.nber.org/papers/w9853

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
July 2003

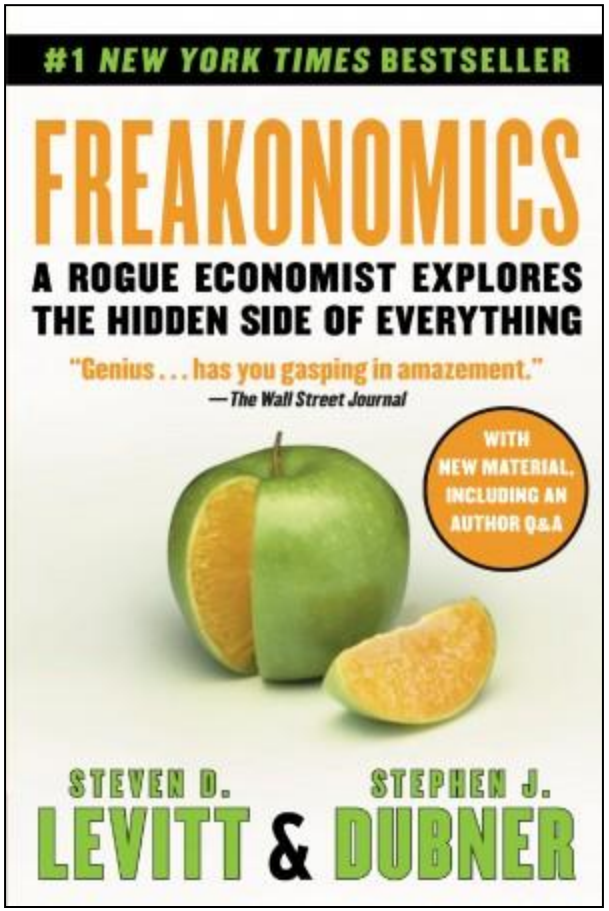
Hamermesh, Daniel S. and Amy M. Parker (2005) "Beauty in the Classroom: Instructors' Pulchritude and Putative Pedagogical Productivity", Economics of Education Review, August 2005, pp. 5-16

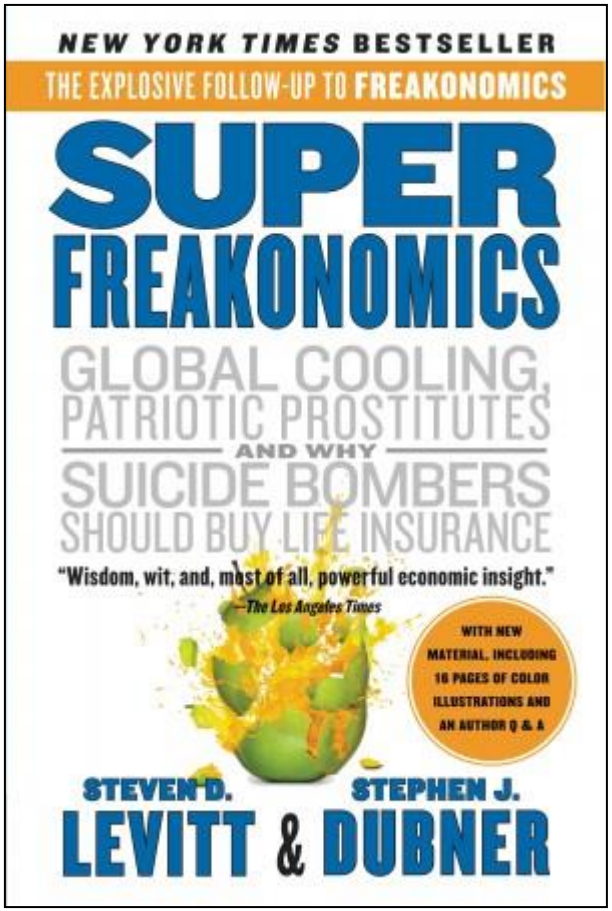
Source: Murtaza Haider "Getting Started with Data Science: Making Sense of Data with Analytics"

Linear regression – statistical perspective

Applications

Regression analysis in Freakonomics books
Econometrics



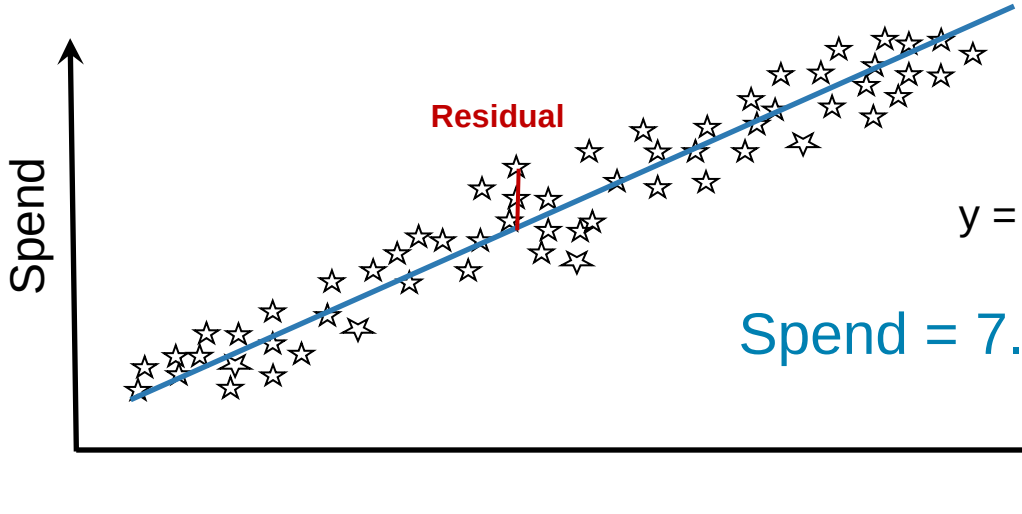


📖 Linear Regression – Data Science Perspective

Linear regression

- Predict a value of a given continuous variable based on the values of other variables, assuming a linear or nonlinear model of dependency
- Virtually endless applications:
  - Election outcomes
  - Future product revenues or commodity prices
  - Wind velocity

✓Both predictive and explanatory power



$y = c_0 + c_1 x_1 + \ldots + c_n x_n$

Spend = 7.4 + 0.37 * Income

Linear regression – usage

Two major categories where regression analysis can be leveraged

1. To predict, estimate or forecast the values: linear regression can be used to fit a predictive model to an observed data set of y and x values
   - The model is fit to a set of known values using “training” data set and validated using a holdout “test” sample
   - Once the model is built it can be leveraged to predict the y values for the records where only x values are available
     - Estimate customer spend (scoring the universe)
     - Estimate future product demand (forecasting)
2. To quantify the strength of the relationship between y and the x_i. To assess which x_j has a strong relationship or whether a particular x_k has a statistically significant relationship with a target variable
   - The model is fit to a set of known values using “training” data set and validated using a holdout “test” sample, then the relationship between the target variable y and predictors x_i is evaluated.
     - Did the patients who received treatment (actual medication) showed statistically significant improvement vs. patients receiving a placebo?
     - What was the biggest driver in customer response (TV ad, banner, direct mail, email)

Estimation / prediction

Prediction example

- Can we predict the CO_2 emission of car without testing it?
- The CO_2 emission of a car is calculated based on the engine size, class, model, make, cylinder, fuel consumption of that car. Prediction is used to predict its expected CO_2 emission.

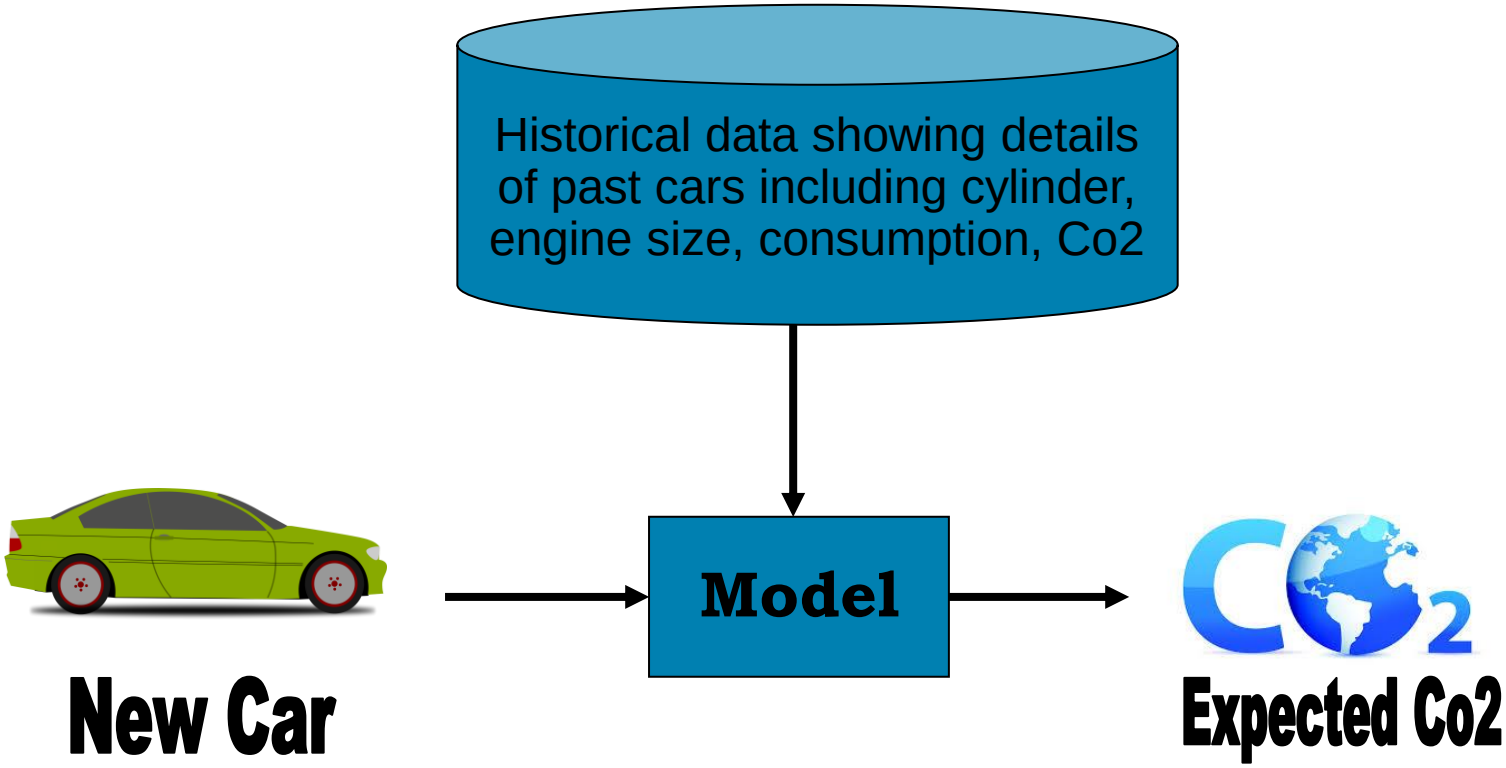
What is a prediction?

Prediction is similar to classification but models are
Continuous / Numerical / Ordered valued

How does it work?

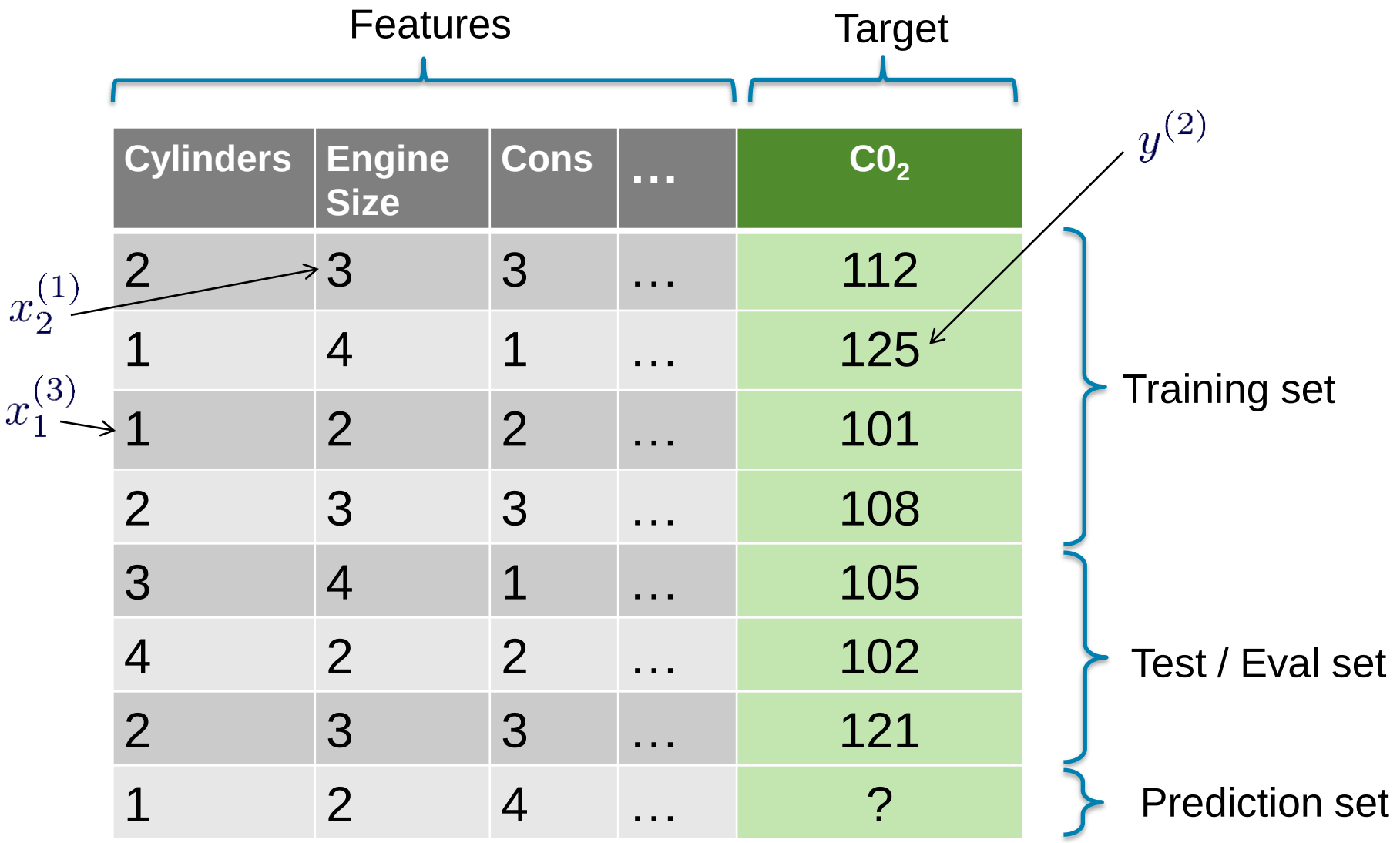
1. Split your data into training and test set
2. Construct a model using training set
3. Evaluate your model using test set
4. Use model to predict unknown value

Estimation / prediction



Predict expected CO_2 emissions for a new car

Dataset


Features
Target
Cylinders
Engine Size
Cons
…
C0$_2$
2 3 3 … 112
1 4 1 … 125
1 2 2 … 101
2 3 3 … 108
3 4 1 … 105
4 2 2 … 102
2 3 3 … 121
1 2 4 … ?
$x_2^{(1)}$
$x_1^{(3)}$
$y^{(2)}$
Training set
Test / Eval set
Prediction set

Prediction

- **Algorithms**:
  - Regression
    - Simple regression
    - Multiple regression
    - Linear regression
    - Non-linear regression
  - *k*-nearest neighbor methods
  - Neural Networks
  - Support Vector Regression

- **What is regression analysis?**
  - Simple regression analysis (one independent variable)
  - Multiple regression analysis (multiple independent variables)

- **What are data science applications**:
  - Marketing: sales forecasting
  - Psychology: satisfaction analysis
  - …

53

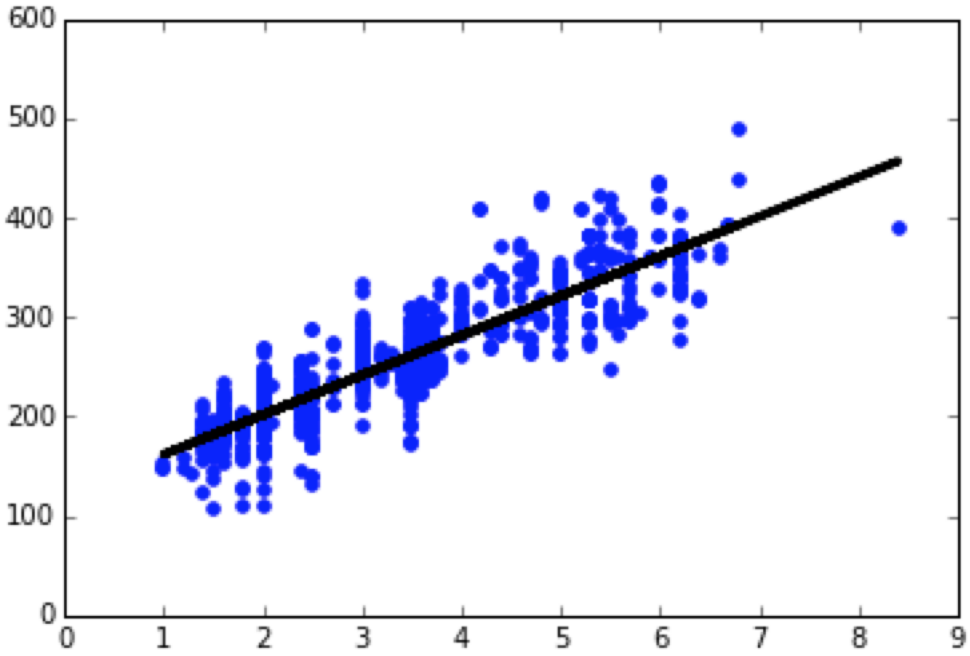
Simple linear regression

- **Target value** is expected to be a **linear combination** of the **input variables** (straight line)
- Regression computes β_i from data to **minimize squared error** to '**fit**' the data

$$y = \beta_0 + \beta_1 x$$

response variable

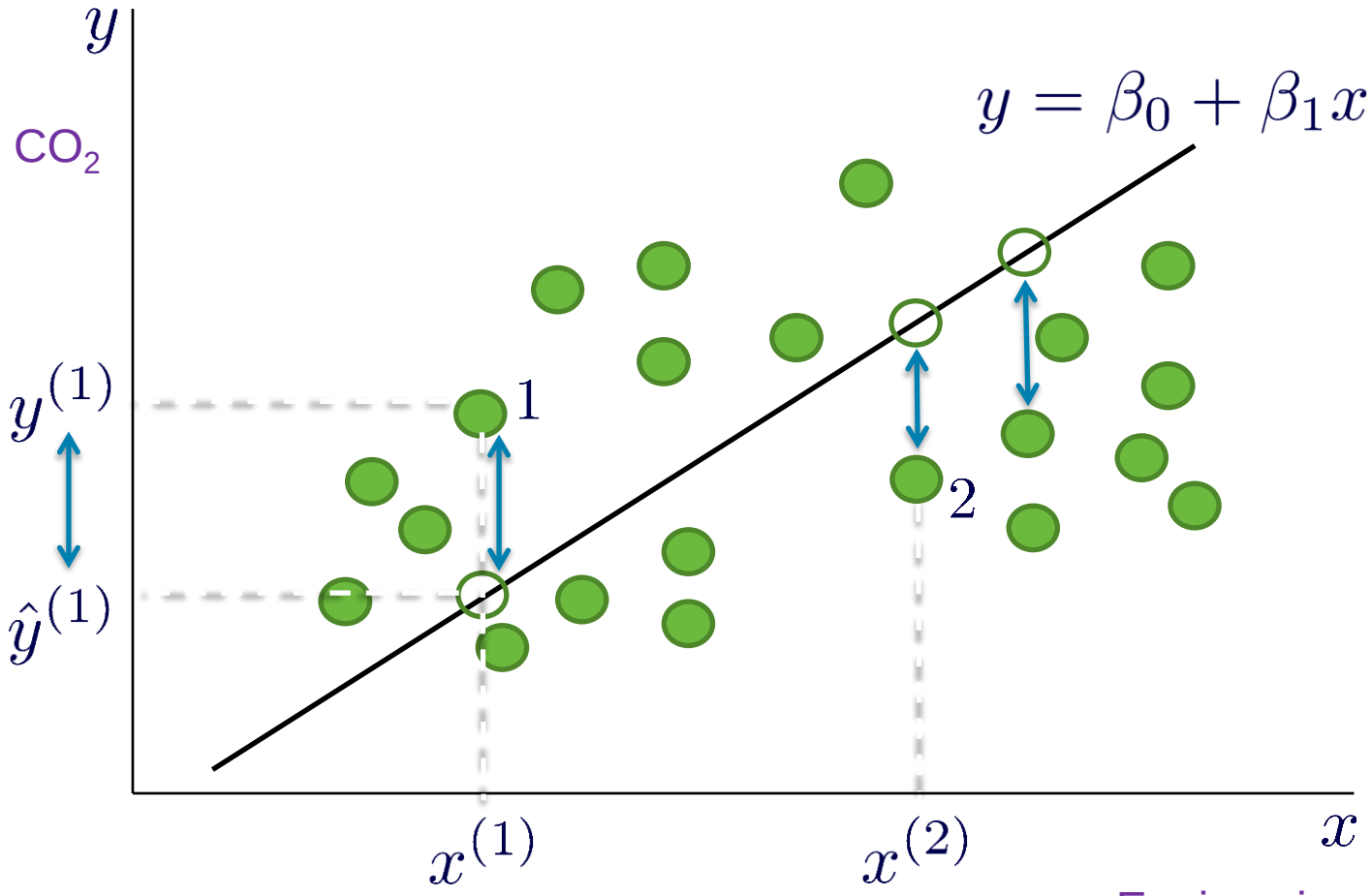
a single predictor



54

Linear regression – ordinary least squares (OLS)

$$\min_{\beta_0,\beta_1} \left(y^{(1)} - \underbrace{(\beta_0 + \beta_1 x^{(1)})}_{\hat{y}^{(1)}}\right)^2 + \left(y^{(2)} - (\beta_0 + \beta_1 x^{(2)})\right)^2 + \dots$$



Multiple linear regression

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- $y \Rightarrow$ dependent / target / predicted / response variable
- x_1, x_2, ... $\Rightarrow$ independent / input / feature / *predictor* variable
- β_1, β_2, ... $\Rightarrow$ coefficients of dependent variables
- $\beta_0 \Rightarrow$ intercept
- $\beta_1 > 0 \Rightarrow$ positive association
- $\beta_1 < 0 \Rightarrow$ negative association
- $\beta_1 = 0 \Rightarrow$ no association

Many nonlinear functions can be transformed into the above form

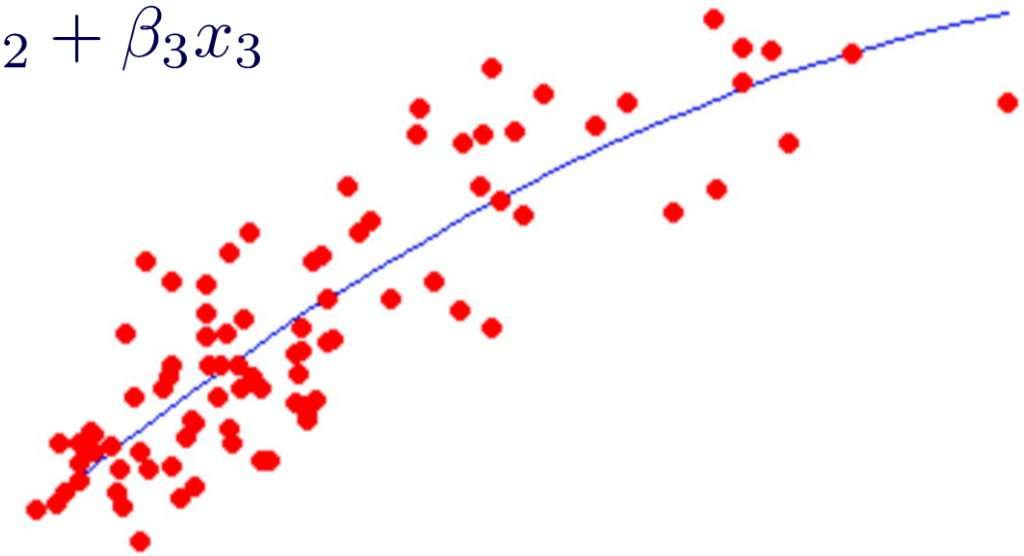
Non-linear regression

- Some nonlinear models can be modeled by a polynomial function
- A polynomial regression model can be transformed into linear regression model
- For example

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

is converted to linear regression with new variables $x_2 = x^2,\ x_3 = x^3$

$$y = \beta_0 + \beta_1 x + \beta_2 x_2 + \beta_3 x_3$$



Accuracy

- Measure predictor accuracy – measure how far off the predicted value ($\hat{y}$) is from the actual known value (y)
- Loss function measures the error between $y^{(j)}$ and the predicted value $\hat{y}^{(j)}$
  - Absolute error $|y^{(j)} - \hat{y}^{(j)}|$
  - Squared error $(y^{(j)} - \hat{y}^{(j)})^2$
- Test error (generalization error) – the average loss over the test set
  - Mean Absolute Error $\frac{\sum_{j=1}^{m} |y^{(j)} - \hat{y}^{(j)}|}{m}$
  - Mean Squared Error (Residual Sum of Squares * *m*) $\frac{\sum_{j=1}^{m} (y^{(j)} - \hat{y}^{(j)})^2}{m}$
  - Relative Absolute Error $\frac{\sum_{j=1}^{m} |y^{(j)} - \hat{y}^{(j)}|}{\sum_{j=1}^{m} |y^{(j)} - \bar{y}|}$
  - Relative Squared Error $\frac{\sum_{j=1}^{m} (y^{(j)} - \hat{y}^{(j)})^2}{\sum_{j=1}^{m} (y^{(j)} - \bar{y})^2}$
  - R^2 = (1 – Relative Squared Error)

Regression analysis: R^2 and variable selection

- Goodness of fit in linear regression models is generally measured by using the R^2
- R^2 measures how well the regression line approximates the real data points, it also portrays percent of variance in the data explained by regression model
  - If the value is close to 1, the model fits perfectly and explains all variance
  - If the value is close to 0, then the model does not fit the data and/or doesn't explain any variance
- Variable preparation:
  - Interval variables can be binned or bucketed in order to capture nonlinear relationship
  - Categorical variables must be converted into binary vectors. Data sample must be large enough to accommodate all degrees of freedom
- **Variable selection** (LASSO algorithm):

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$$

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$$

$$\text{subject to} \quad \|\boldsymbol{\beta}\|_1 \leq \varepsilon$$

📖 Linear Regression – Machine Learning Perspective

Linear regression – data mining and machine learning perspective

- **Input** (features):
  - n - number of features
  - $x^{(j)}$ - input (features) of *j*-th training example
  - $x_i^{(j)}$ - value of feature *i* in *j*-th training example

$$\boldsymbol{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- **Hypothesis**:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x} = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \quad (x_0 = 1)$$

- **Parameters**:

$$\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)^T$$

- **Cost function**:

$$J(\theta_0, \theta_1, \theta_2, \ldots, \theta_n) = \frac{1}{2m} \sum_{j=1}^{m} \left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)}) - y^{(j)} \right)^2$$

- **Optimization**:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Linear regression – data mining and machine learning perspective

- **Optimization**:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

- **Solution algorithms**:
  - Non-linear optimization methods, e.g., iterative algorithms such as gradient descent algorithms, Newton and Quasi-Newton algorithms, etc.
  - Linear algebra (normal equations), i.e., solving $\nabla J(\boldsymbol{\theta}) = 0$

$$\nabla J(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_0} \\ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} \\ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_2} \\ \vdots \\ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_n} \end{pmatrix}$$

- **Solving normal equations**:

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_i} = \frac{1}{m} \sum_{j=1}^{m} \left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)}) - y^{(j)} \right) x_i^{(j)} = 0$$

- **Solve system of linear equations for $\boldsymbol{\theta}$**:

$$\sum_{j=1}^{m} \left(\boldsymbol{\theta}^T \boldsymbol{x}^{(j)} - y^{(j)} \right) x_i^{(j)} = 0$$

Linear regression – data mining and machine learning perspective

- **Solving normal equations** (***m*** examples, ***n*** features):

$$\sum_{j=1}^{m} \boldsymbol{\theta}^T \boldsymbol{x}^{(j)} \cdot x_i^{(j)} = \sum_{j=1}^{m} y^{(j)} \cdot x_i^{(j)} \qquad i = 1, \ldots, n$$

- **Notation**:

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \cdots & x_n^{(2)} \\ 1 & x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \cdots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \cdots & x_n^{(m)} \end{pmatrix} \qquad \boldsymbol{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

- **Solution**:

$$\boldsymbol{\theta} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

- **Solution algorithm** (solve system of linear equations with unknown $\boldsymbol{\theta}$):

$$(\boldsymbol{X}^T \boldsymbol{X}) \cdot \boldsymbol{\theta} = \boldsymbol{X}^T \boldsymbol{y}$$

📖 Logistic Regression – Machine Learning Perspective

Other types of regression analysis

Quantile regression

- Ordinary least squares regression approximates the conditional mean of the response variable, while quantile regression is estimating either the conditional median or other quantiles of the response variable
- This is very helpful in case of skewed data (e.g., income distribution in the US) or to deal with data without suppressing outliers

Logistic regression

- Logistic regression is used to predict categorical target variable
- Most often a variable with a binary outcome
  - Logit and Probit regressions can also be used to predict binary outcome. While the underlying distributions are different, all three models will produce rather similar outcomes
- It is frequently used to estimate the probability of an event
  - Bank customer defaulting on the loan
  - Customer responding to a marketing promotion
  - Spam or not-spam email
  - Malignant or benign tumor

Classification

- **Dataset**: $D = \left\{ (y^{(1)}, \boldsymbol{x}^{(1)}), \ldots, (y^{(m)}, \boldsymbol{x}^{(m)}) \right\}$
- **Targets**: y
- **Features**: $\boldsymbol{x}$
- **Parameters**: $\boldsymbol{\theta}$
- **Classifier**: $h(\boldsymbol{x}|\boldsymbol{\theta}) = h_{\boldsymbol{\theta}}(\boldsymbol{x})$
- **Predictions**: $\hat{y} = h(\boldsymbol{x}|\boldsymbol{\theta})$

Iris setosa (A)

$y = 0$

?

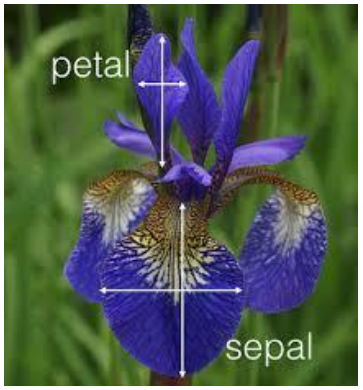
Iris versicolor (B)

$y = 1$

- In **classification** we are trying to **predict discrete targets**
- In the **two-class** problem $y \in \{0, 1\}$ means that y can be equal **0** (“negative class”, e.g., spam) or **1** (“positive class”, e.g., not spam)
- Example classification problem – **classify different flowers using measurements of the flower**

Classification

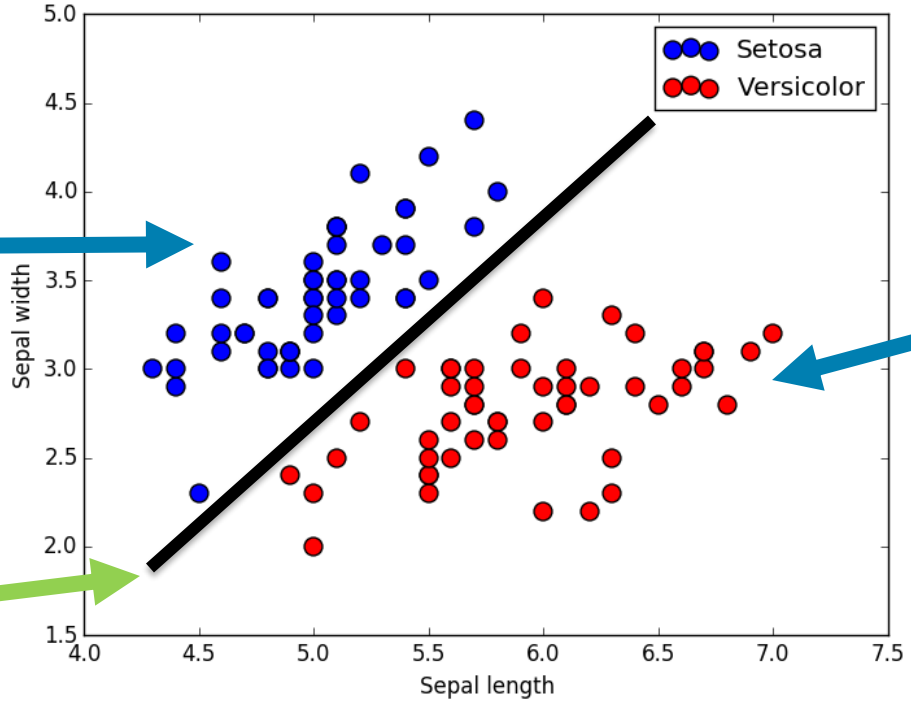
- **Features** are **numerical attributes**
- **Good features** can be used to **predict targets** (different classes)



Fisher's Iris Data				
Sepal length	Sepal width	Petal length	Petal width	Species
x_1	x_2	x_3	x_4	y

- Sometimes we can **separate** different classes with a **linear decision boundary**

A

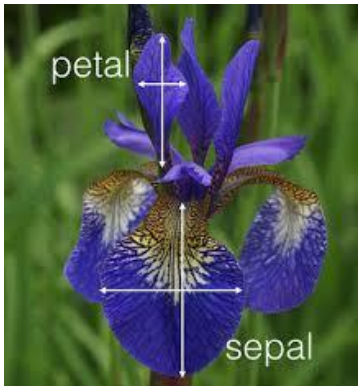


B

decision boundary

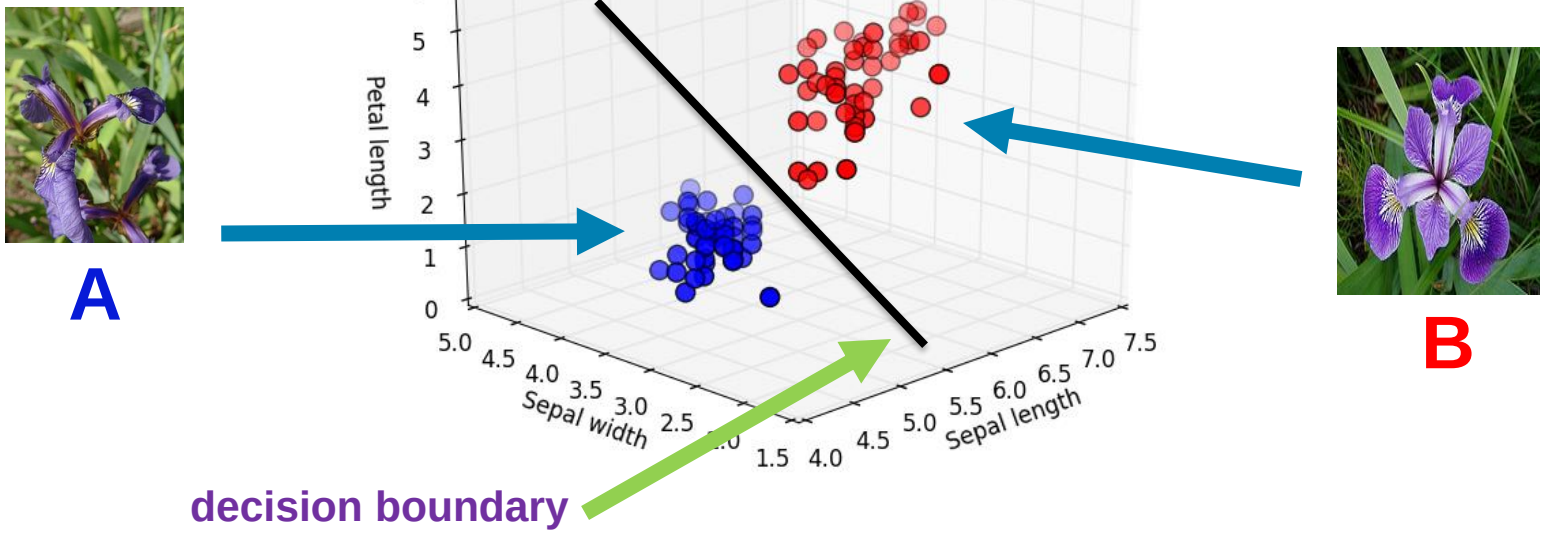
Classification

- **Features** are **numerical attributes**
- **Good features** can be used to **predict targets** (different classes)



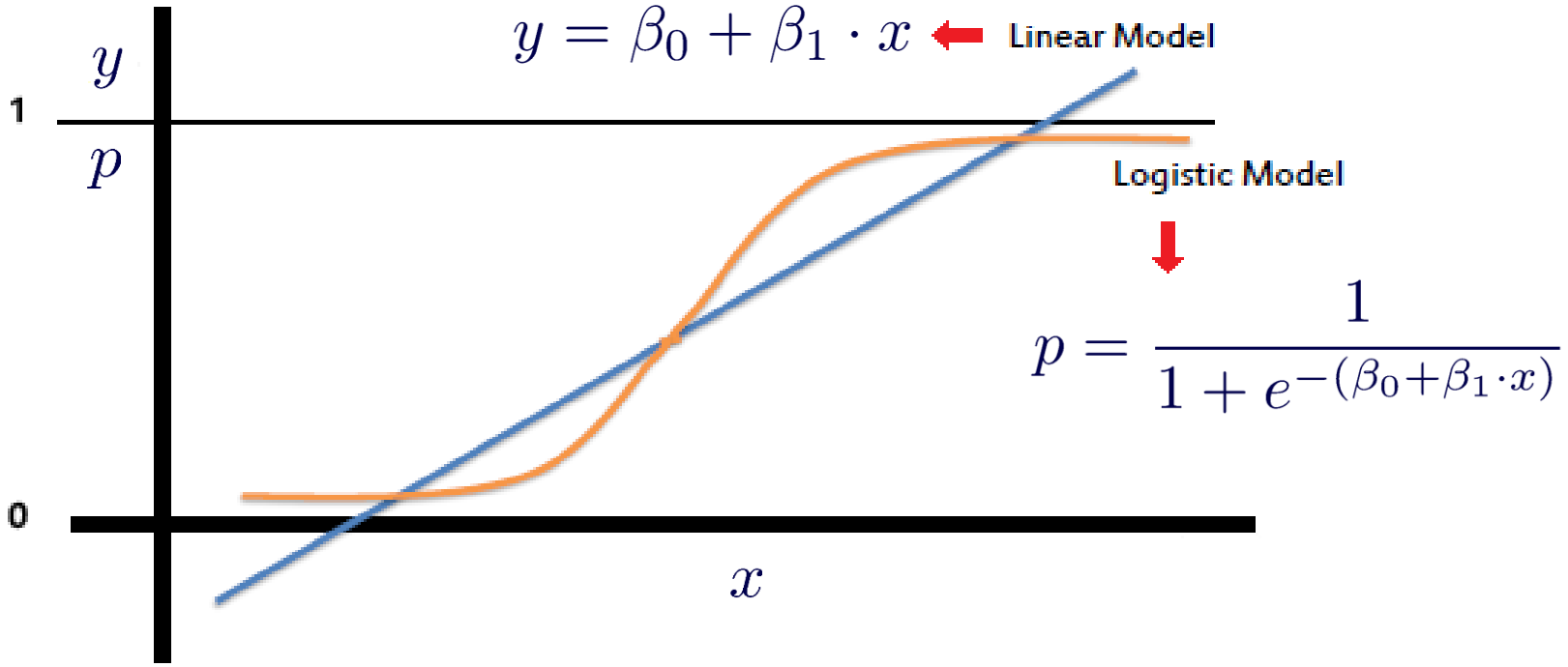
Fisher's Iris Data				
Sepal length	Sepal width	Petal length	Petal width	Species
x_1	x_2	x_3	x_4	y

- Sometimes we can **separate** different classes with a **linear decision boundary**
- **Usually more features can give better performance**



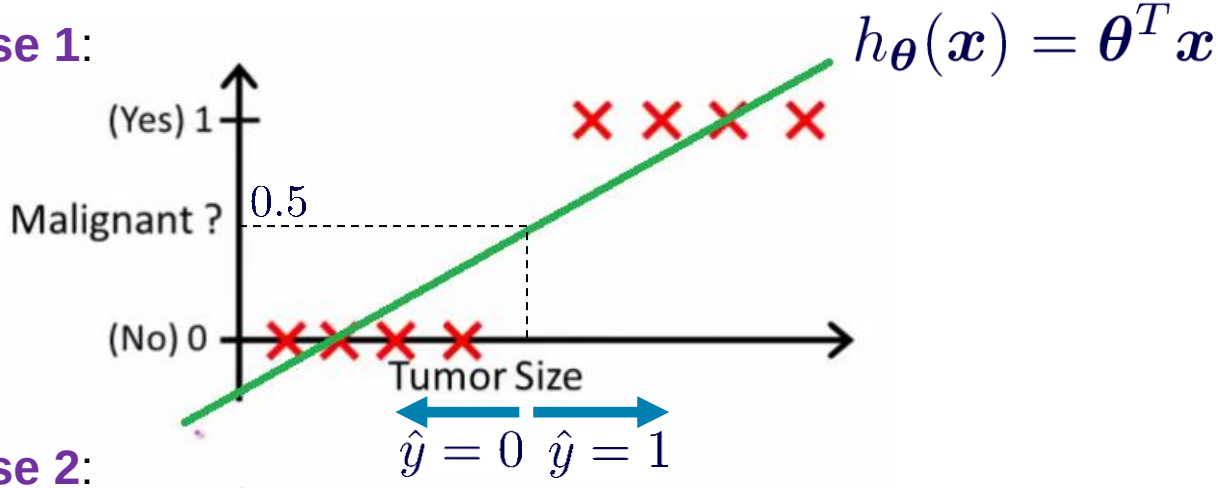
Linear regression vs. logistic regression

- **Linear regression** predicts **real values**
- **Logistic regression** predicts **values in the range of 0 to 1**

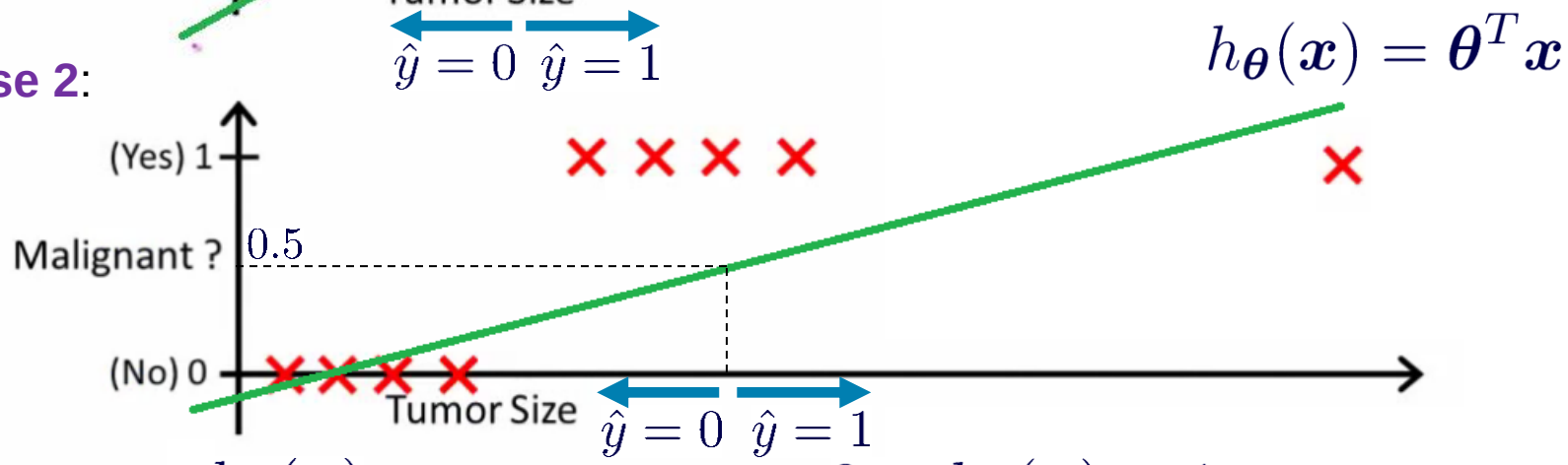


Using linear regression for classification

- **Threshold** classifier output $h_\theta(x)$ at **0.5**:
  - If $h_\theta(x) \geq 0.5$ predict "$y = 1$"
  - If $h_\theta(x) < 0.5$ predict "$y = 0$"
- **Case 1**:



- **Case 2**:



- What to do if $h_\theta(x) < 0$ or > 1? **We want** $0 \leq h_\theta(x) \leq 1$

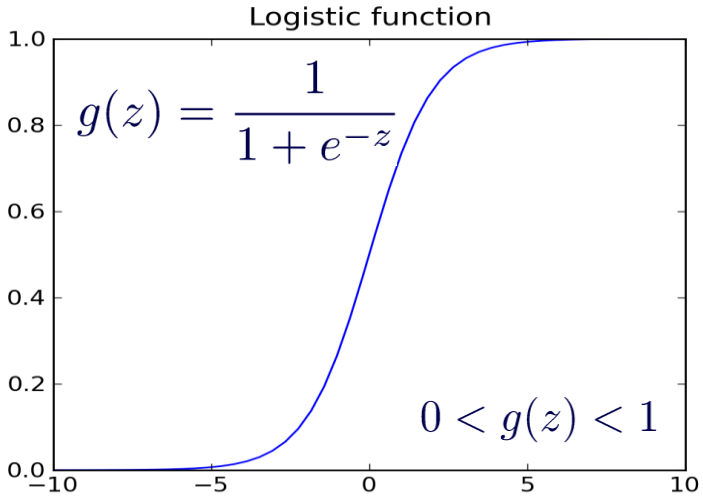
Source: Andrew Ng "Machine Learning"

Logistic regression – hypothesis representation

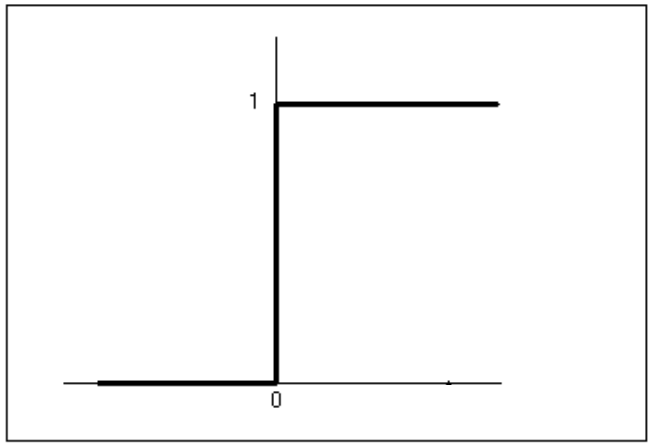
- **We want** $0 \leq h_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$:
  - hypothesis for linear regression $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}$
  - hypothesis for logistic regression $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x})$

- **Sigmoid function / logistic function** $g(z) = \dfrac{1}{1+e^{-z}}$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$



use **logistic function** to approximate **threshold function**



one important reason is that **logistic function** is **differentiable**

- **We need to fit parameters** $\boldsymbol{\theta}$

Source: Andrew Ng “Machine Learning”

Logistic regression – hypothesis representation

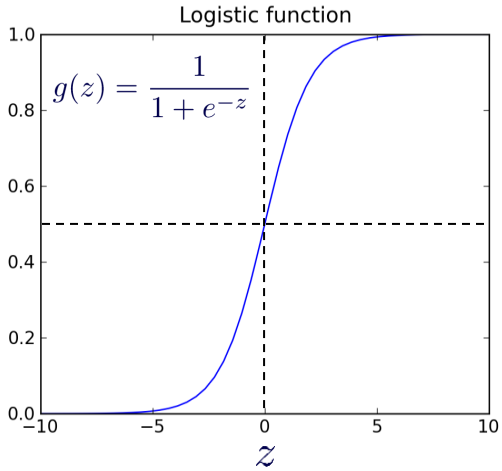
- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) =$ estimated probability that $y = 1$ on input $\boldsymbol{x}$
  - if $\boldsymbol{x} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ \text{tumorSize} \end{pmatrix}$ and $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0.7$
  - tell a patient that 70% chance of tumor being malignant

- Probability that $y = 1$, given $\boldsymbol{x}$, parameterized by $\boldsymbol{\theta}$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = \text{prob}(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta})$$

$$\text{prob}(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) + \text{prob}(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 1$$

- Suppose:
  - predict "$y = 1$" if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0.5$
    → $\boldsymbol{\theta}^T \boldsymbol{x} \geq 0$
  - predict "$y = 0$" if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0.5$
    → $\boldsymbol{\theta}^T \boldsymbol{x} < 0$



$g(z) \geq 0.5$ when $z \geq 0$
$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) \geq 0.5$ when $\boldsymbol{\theta}^T \boldsymbol{x} \geq 0$

$g(z) < 0.5$ when $z < 0$
$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) < 0.5$ when $\boldsymbol{\theta}^T \boldsymbol{x} < 0$

Source: Andrew Ng "Machine Learning"

Logistic regression – decision boundary

- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
  - predict "$y = 1$" if $\boldsymbol{\theta}^T \boldsymbol{x} = -3 + x_1 + x_2 \geq 0$
  - predict "$y = 0$" if $\boldsymbol{\theta}^T \boldsymbol{x} = -3 + x_1 + x_2 < 0$

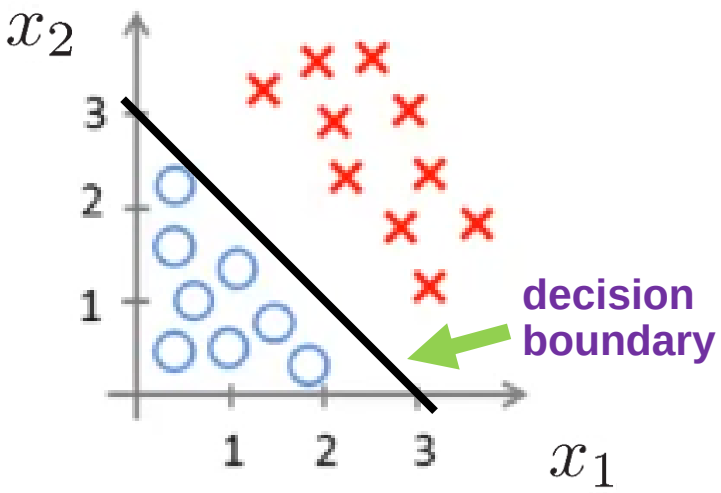
fit

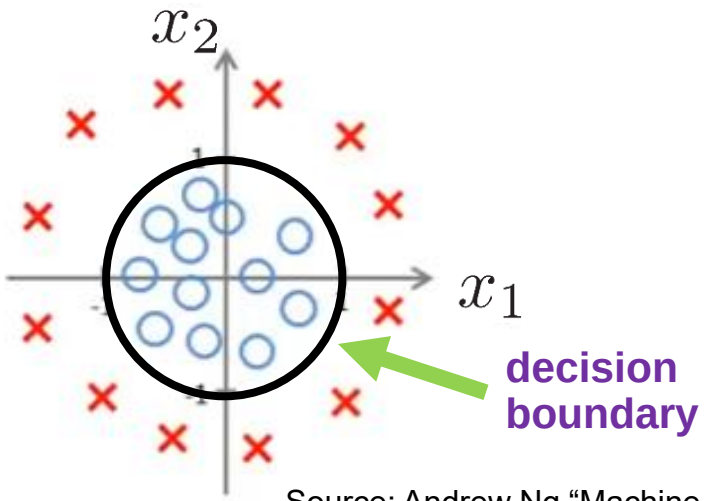
$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix}$$

- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$
  - predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$
  - predict "$y = 0$" if $-1 + x_1^2 + x_2^2 < 0$

fit

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$





Source: Andrew Ng "Machine Learning"

Logistic regression – decision boundary

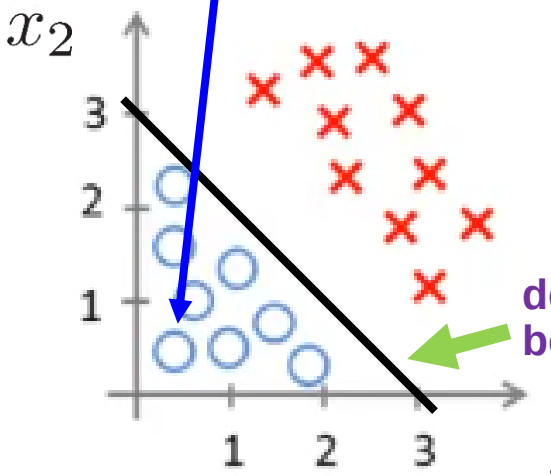
- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
  - predict "$y = 1$" if $\boldsymbol{\theta}^T \boldsymbol{x} = -3 + x_1 + x_2 \geq 0$
  - predict "$y = 0$" if $\boldsymbol{\theta}^T \boldsymbol{x} = -3 + x_1 + x_2 < 0$

fit

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix}$$

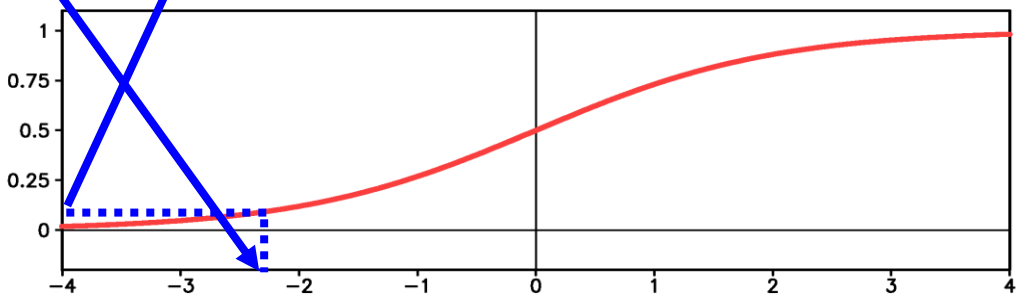
$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} = \frac{1}{1 + e^{-(-3 + x_1 + x_2)}}$$

$$h_{\boldsymbol{\theta}}(0.4, 0.4) = \frac{1}{1 + e^{-\underbrace{(-3+0.4+0.4)}_{-2.2}}} = 0.1$$



decision boundary

$-3 + x_1 + x_2 = 0$



Source: Andrew Ng "Machine Learning"

Logistic regression – decision boundary

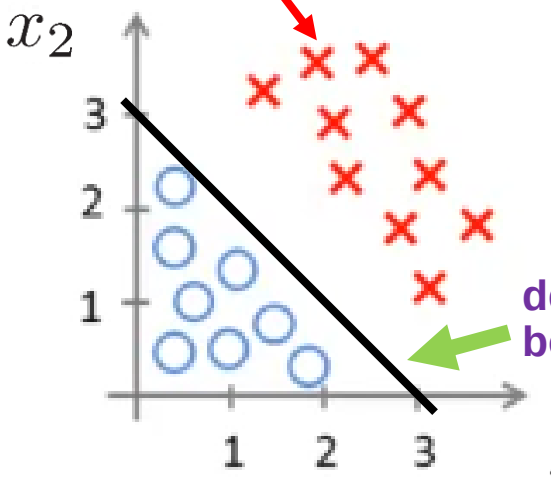
- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
  - predict "$y = 1$" if $\boldsymbol{\theta}^T \boldsymbol{x} = -3 + x_1 + x_2 \geq 0$
  - predict "$y = 0$" if $\boldsymbol{\theta}^T \boldsymbol{x} = -3 + x_1 + x_2 < 0$

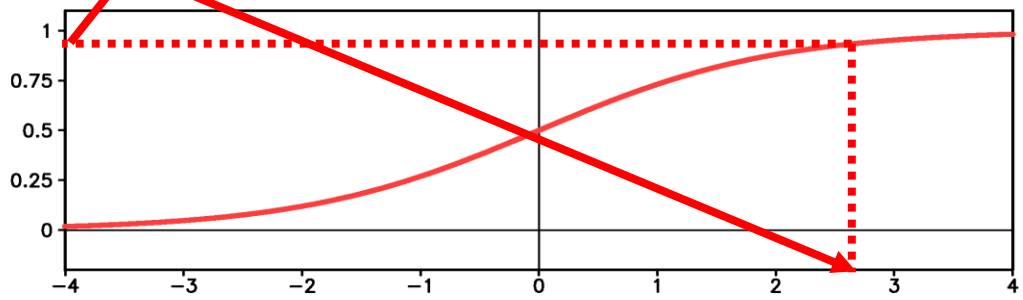
fit

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix}$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} = \frac{1}{1 + e^{-(-3 + x_1 + x_2)}}$$

$$h_{\boldsymbol{\theta}}(2.0, 3.6) = \frac{1}{1 + e^{-\underbrace{(-3+2.0+3.6)}_{2.6}}} = 0.93$$





decision boundary

x_1

$-3 + x_1 + x_2 = 0$

Source: Andrew Ng "Machine Learning"

Logistic regression – decision boundary

- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$
  - predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$
  - predict "$y = 0$" if $-1 + x_1^2 + x_2^2 < 0$

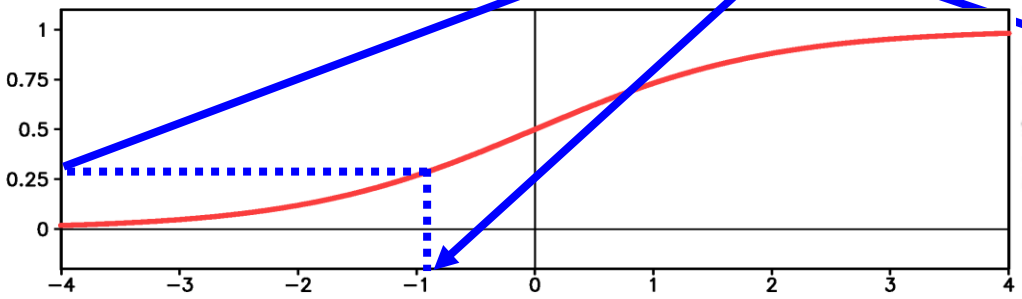
fit

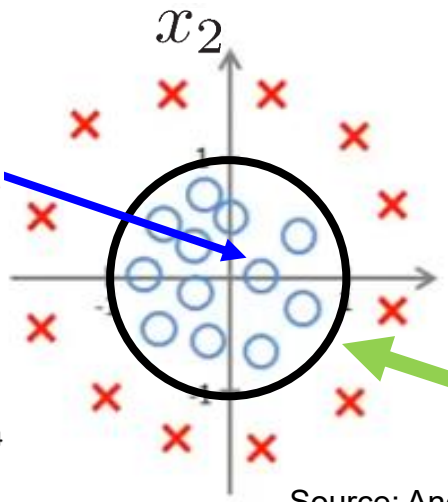
$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} = \frac{1}{1 + e^{-(-1 + x_1^2 + x_2^2)}}$$

$$h_{\boldsymbol{\theta}}(0.3, 0) = \frac{1}{1 + e^{-(-1 + 0.3^2 + 0.0^2)}} = 0.3$$

-0.91





$x_1^2 + x_2^2 = 1$

decision boundary

Source: Andrew Ng "Machine Learning"

Logistic regression – decision boundary

- $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$
  - predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$
  - predict "$y = 0$" if $-1 + x_1^2 + x_2^2 < 0$

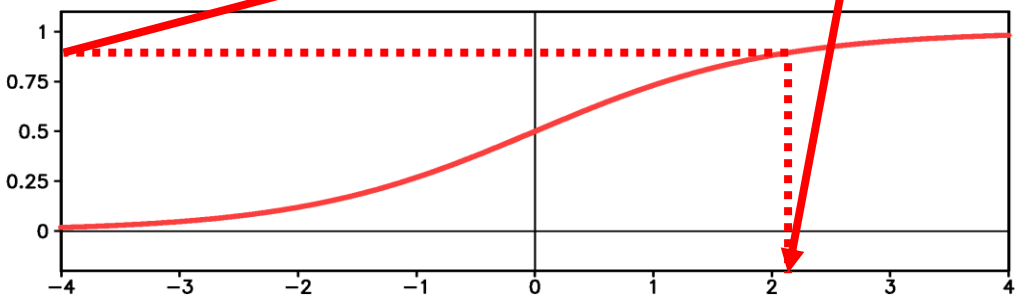
fit

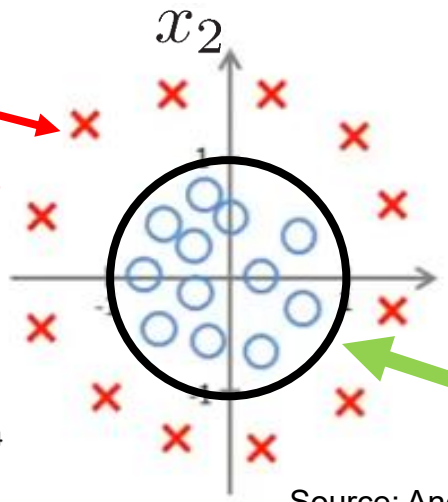
$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} = \frac{1}{1 + e^{-(-1 + x_1^2 + x_2^2)}}$$

$$h_{\boldsymbol{\theta}}(1.2, 1.3) = \frac{1}{1 + e^{-(-1 + 1.2^2 + 1.3^2)}} = 0.9$$

2.1





$x_1^2 + x_2^2 = 1$

decision boundary

Source: Andrew Ng "Machine Learning"

Logistic regression – cost function

- **Input** (training set): $D = \left\{ (y^{(1)}, \boldsymbol{x}^{(1)}), \ldots, (y^{(m)}, \boldsymbol{x}^{(m)}) \right\}$
  - n - number of features
  - m - number of examples
  - $x^{(j)}$ - input (features) of *j*-th training example
  - $x_i^{(j)}$ - value of feature *i* in *j*-th training example

$$y \in \{0, 1\} \qquad \boldsymbol{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad x_0 = 1$$

- **Hypothesis**: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \dfrac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$
- **Parameters**: $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)^T$
- **Cost function**: $J(\boldsymbol{\theta}) = \dfrac{1}{m} \sum_{j=1}^{m} \underbrace{\text{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)}),\ y^{(j)} \right)}_{\frac{1}{2}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)}) - y^{(j)} \right)^2}$

cost function that we used for linear regression is now **non-convex**

- **Optimization**: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

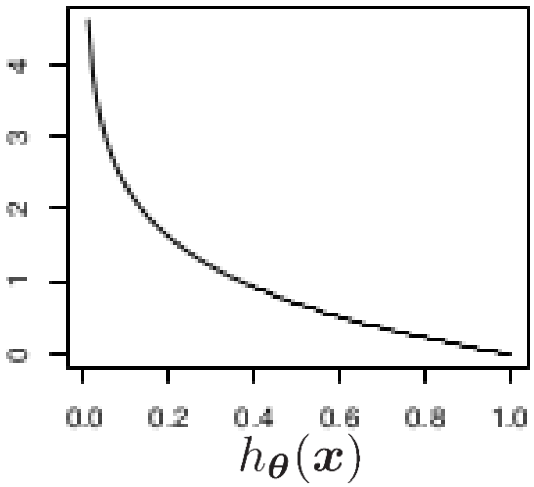
Source: Andrew Ng “Machine Learning”

Logistic regression – cost function

- **Cost function**:

$$\text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}),\ y) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } \ y = 1 \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } \ y = 0 \end{cases}$$

if $y = 1$

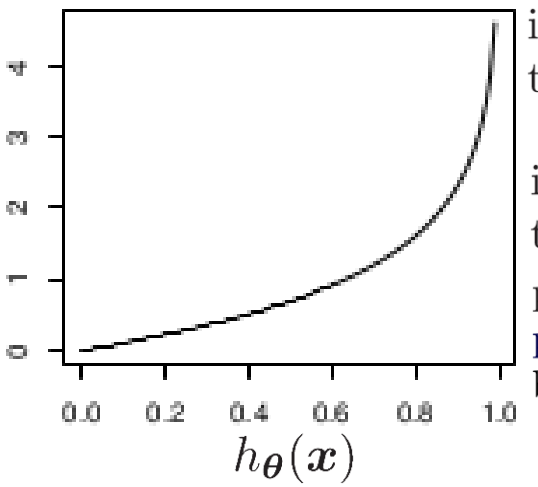


if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 1$
then cost $= 0$

if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 0$
then cost $\to \infty$

predicted
$\text{prob}(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 0$
but $y = 1$

if $y = 0$



if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$
then cost $= 0$

if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 1$
then cost $\to \infty$

predicted
$\text{prob}(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 0$
but $y = 0$

- **More compact form of cost function**:

$$\text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}),\ y) = -y \log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) - (1 - y) \log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}))$$

$$\text{if } y = 1: \quad \text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x}))$$
$$\text{if } y = 0: \quad \text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}))$$

Source: Andrew Ng “Machine Learning”

Logistic regression – optimization problem

- **Cost function**:

$$J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{j=1}^{m} \text{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)}),\ y^{(j)}\right)$$

in statistics this **cost function** can be derived using principle of **Maximum Likelihood estimation**

$$= -\frac{1}{m}\sum_{j=1}^{m}\left[y^{(j)}\log(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)})) + (1-y^{(j)})\log(1-h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)}))\right]$$

- **Optimization** to fit parameters $\boldsymbol{\theta}$: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- **Solution algorithms**:
  - Non-linear optimization methods, e.g., iterative algorithms such as gradient descent algorithms, Newton and Quasi-Newton algorithms

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_i} = \frac{1}{m}\sum_{j=1}^{m}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)}) - y^{(j)}\right)x_i^{(j)}$$

$$\nabla J(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_0} \\ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1} \\ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_2} \\ \vdots \\ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_n} \end{pmatrix}$$

- To make **predictions** given new $\boldsymbol{x}$ we **output** $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}}$

Logistic regression – multiclass classification

- **One-vs-all (one-vs-rest)**
  - Train a logistic regression classifier $h_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{x})$ for each class *k* to predict the probability that $y = k$
  - On a new input ***x***, to make a prediction, pick the class *k* that maximizes

$$\max_k \; h_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{x})$$

- **Email tagging – Primary, Social, Promotions, Forums**

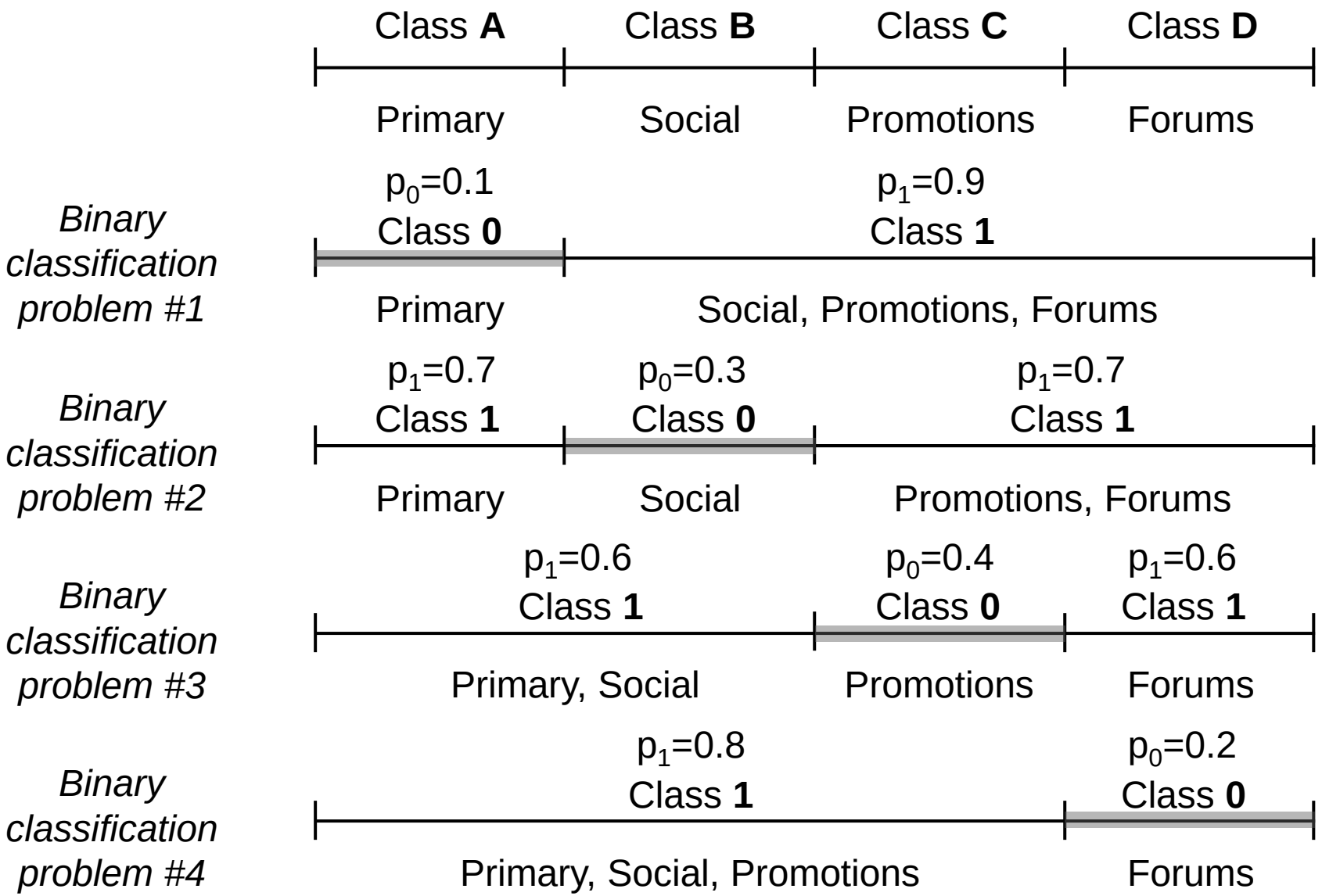
- **Classify different flowers using measurements of the flower**

Iris setosa (A)

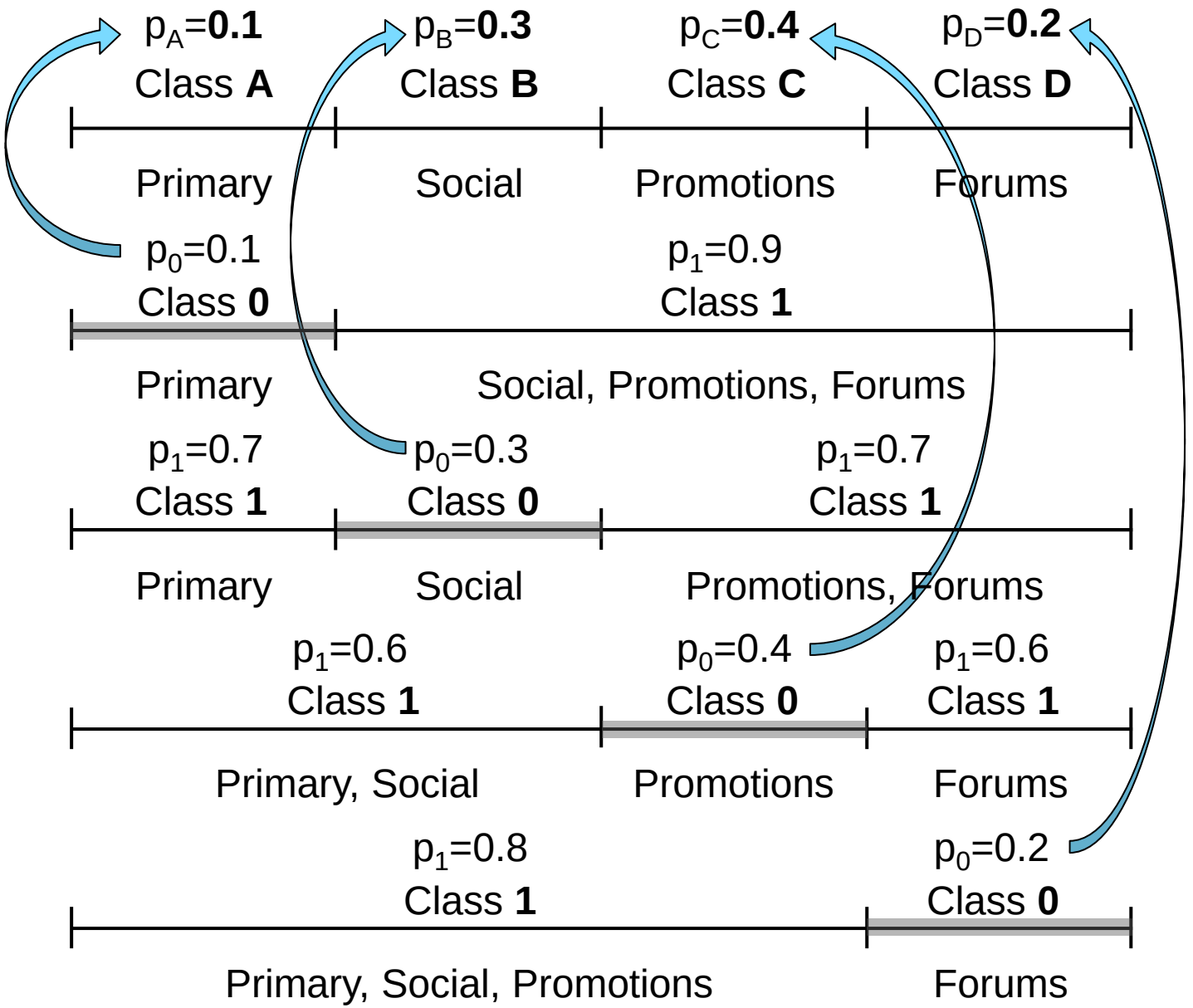
Iris versicolor (B)

Iris virginica (C)

Multi-class classification algorithm (one-vs-all) – e-mail tagging

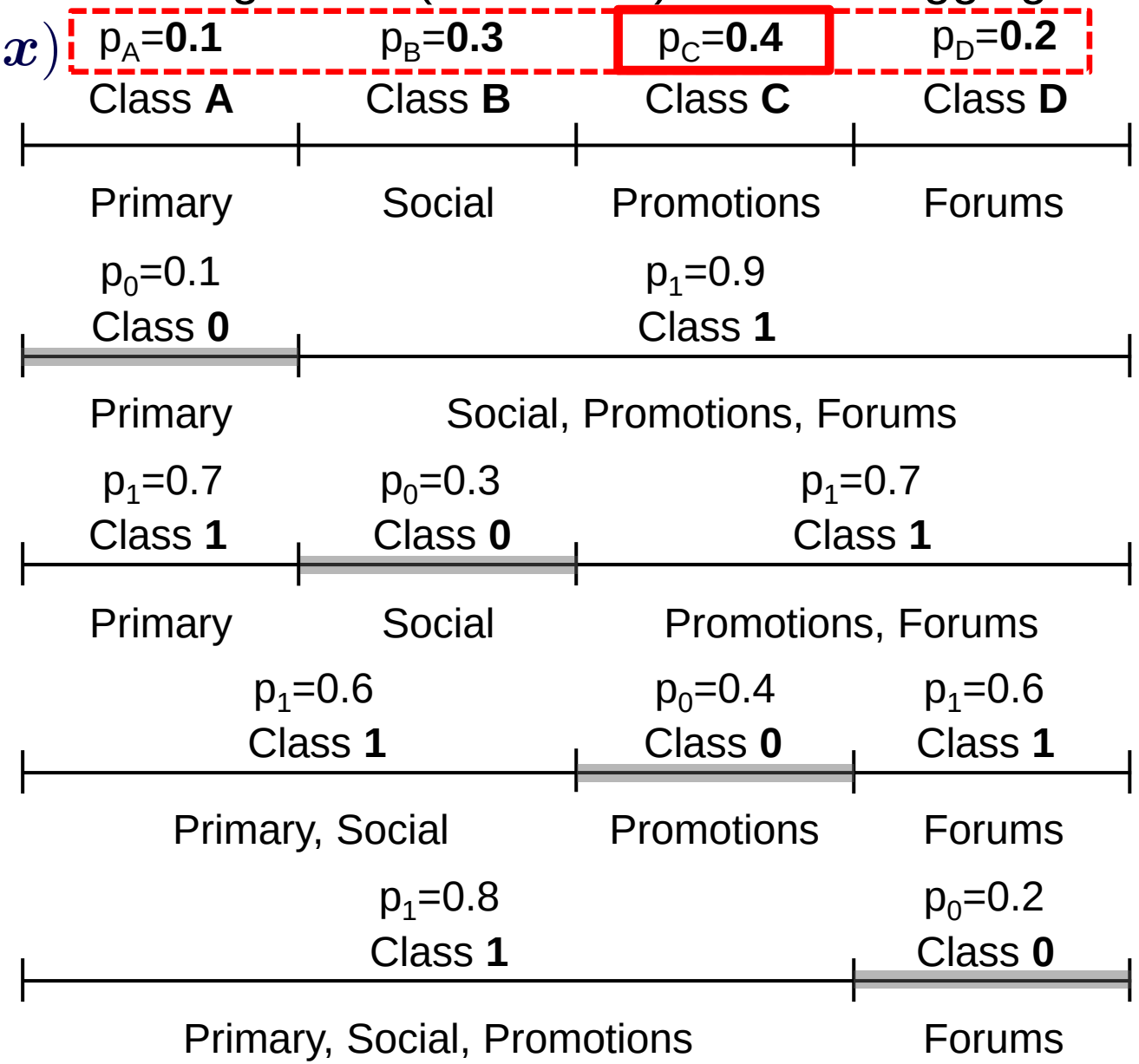


Multi-class classification algorithm (one-vs-all) – e-mail tagging

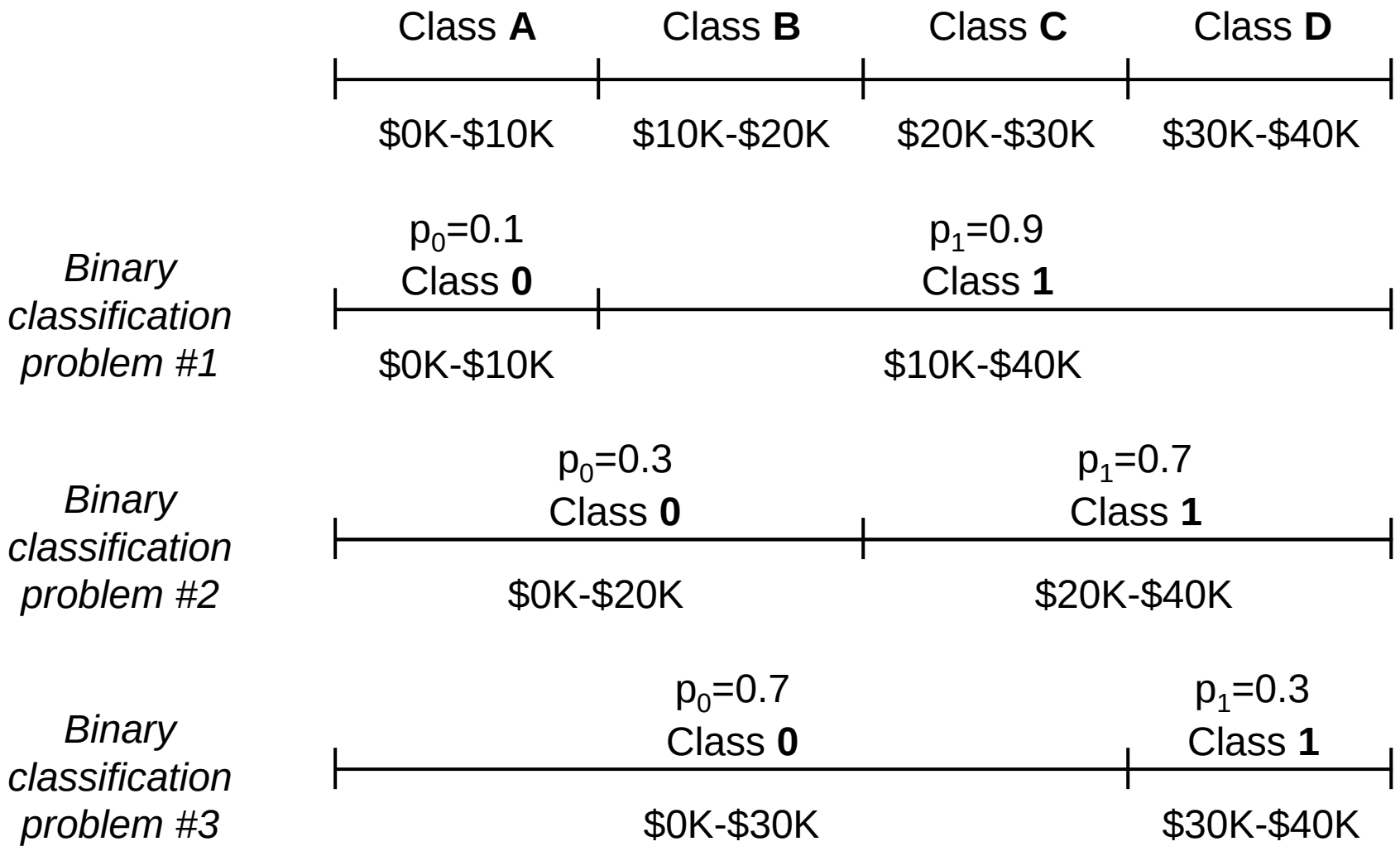


Multi-class classification algorithm (one-vs-all) – e-mail tagging

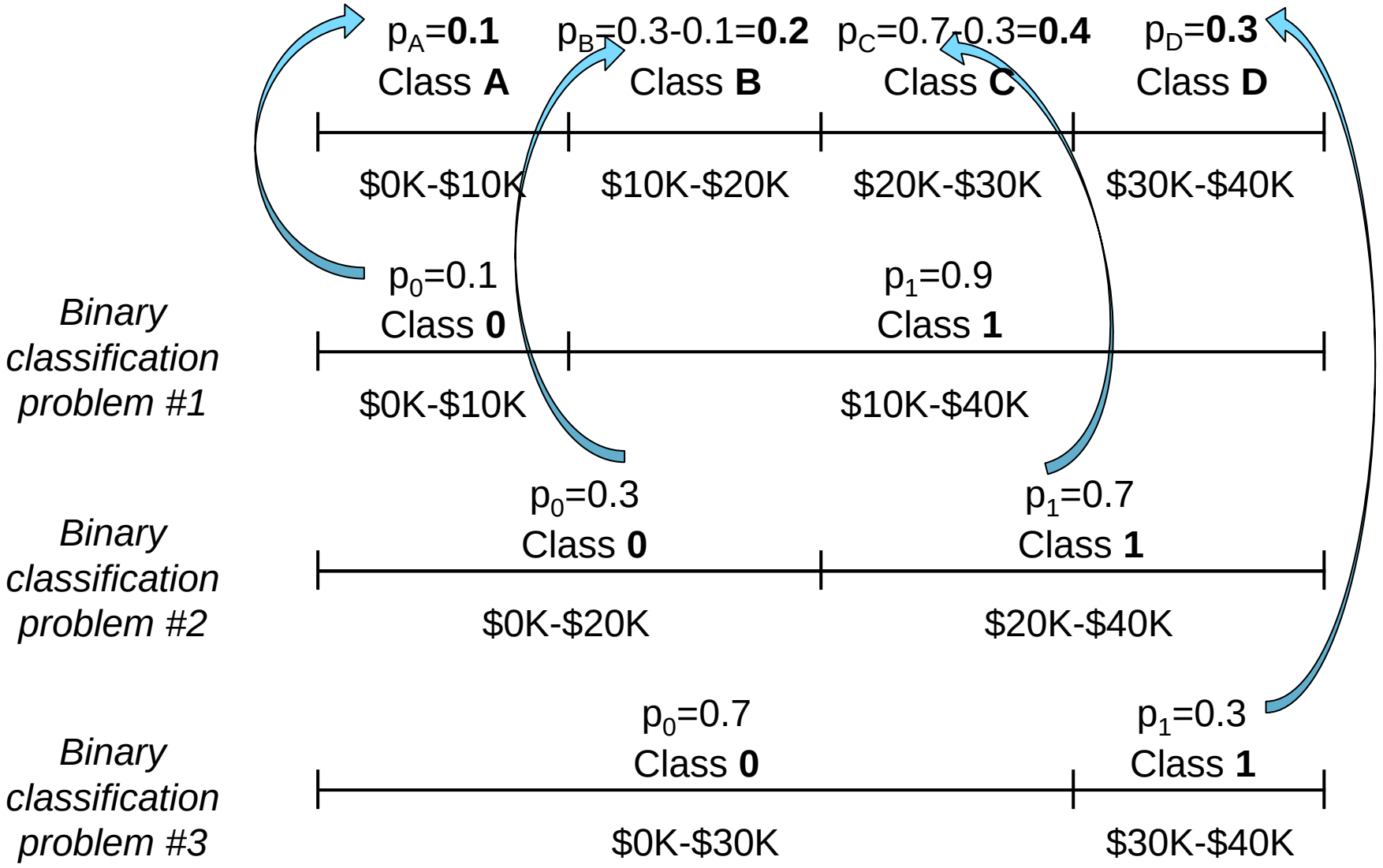
$$\max_{k=A,B,C,D} h_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{x})$$



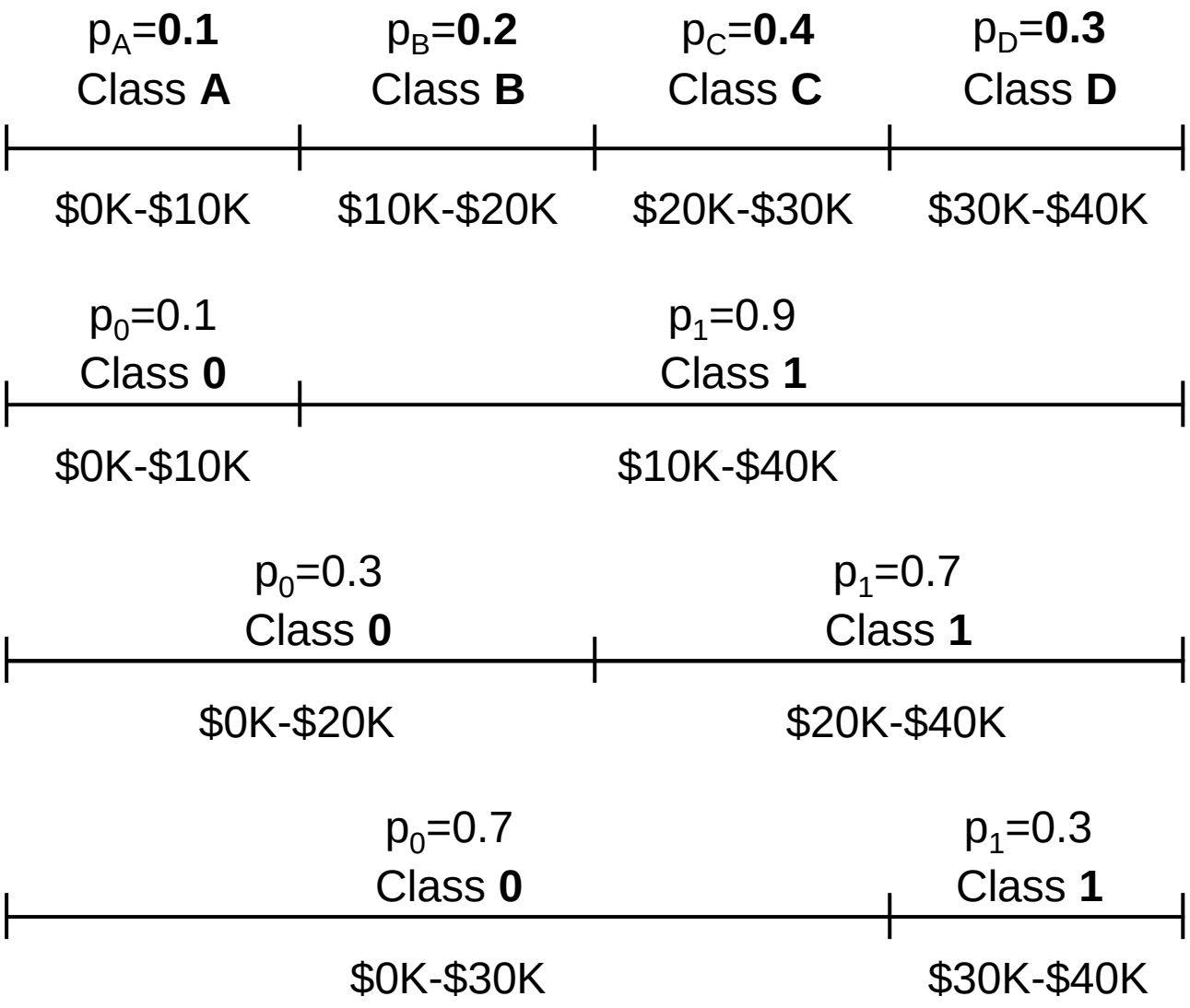
Ordinary multi-class classification algorithm – salary classification



Ordinary multi-class classification algorithm – salary classification

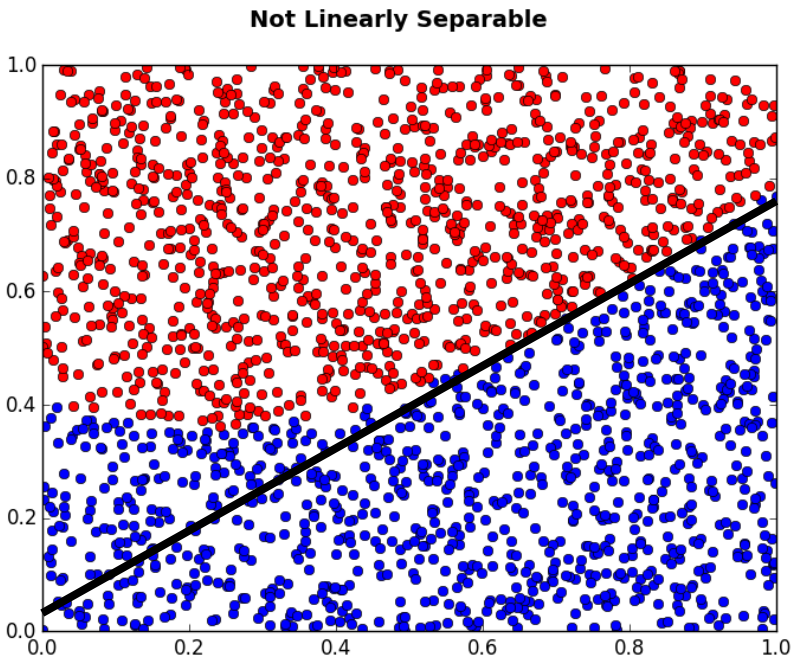


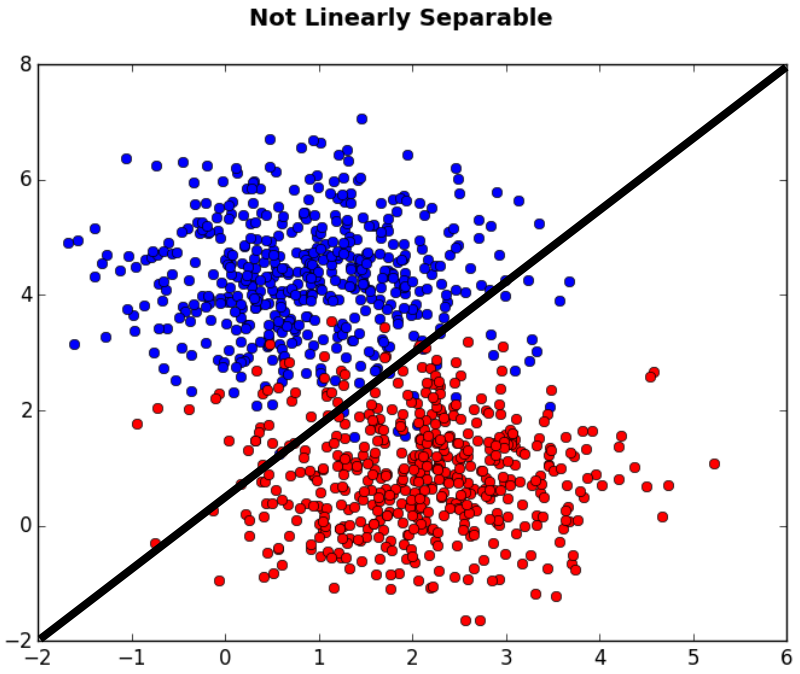
Ordinary multi-class classification algorithm – salary classification



Classification – non linearly separable classes

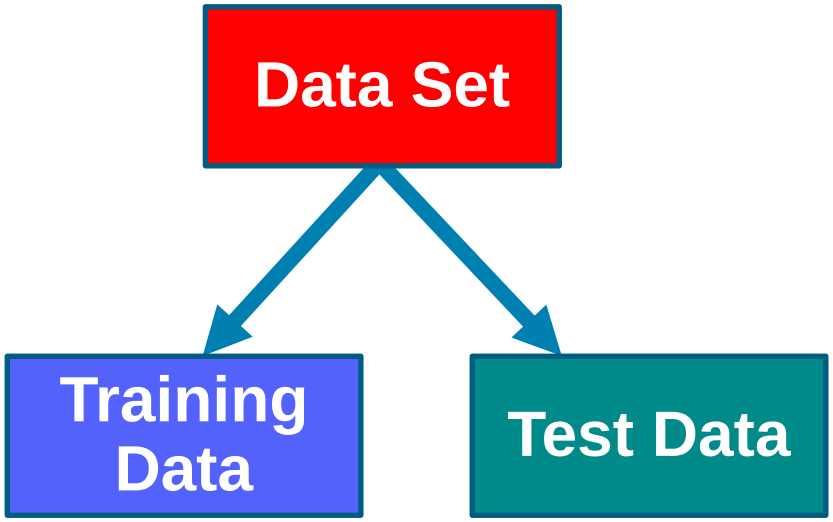
- Sometimes we can **not separate** data **linearly**

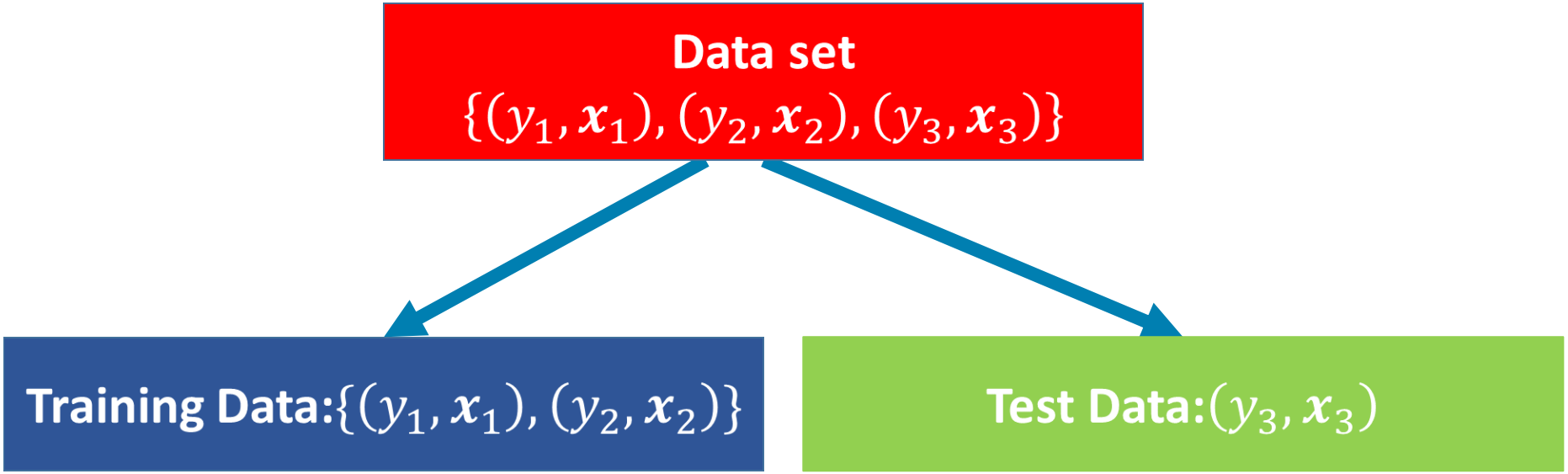




Training and testing data

- First train the system using one part of the data set
- Then test the system using another part of the data set





Overfitting

- **In many models it can be shown that:**

- **A model generalizes well if test error is similar to training error:**

Evaluation method

- **Holdout method**
  - Given data is randomly partitioned into two independent sets
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Random sampling: a variation of holdout
    - Repeat holdout *k* times, accuracy = avg. of the accuracies obtained

- **Cross-validation** (*k*-fold, where *k* = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - Leave-one-out: *k* folds where *k* = # of tuples, for small sized data
  - Stratified cross-validation: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

Bias-Variance Tradeoff and Hyperparameter Tuning

Linear regression – least squares

- Linear regression fits a linear model with coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots)^T$ to minimize the residual sum of squares between the observed responses ($\boldsymbol{y}$) in the dataset, and the responses predicted by the linear approximation ($\hat{\boldsymbol{y}}$)
- Mathematically it solves a problem of the form:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$$

- Method of least squares estimates the best-fitting straight line

- We discussed earlier how to compute $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots)^T$ that minimizes the residual sum of squares

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

Bias-variance tradeoff

- Because $\boldsymbol{X}\hat{\boldsymbol{\beta}}$ fits our data well, this doesn't mean that it will be a good fit to new data
- Prediction error for new data $\|\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{y}}\|_2^2$ should be small:

$$\mathrm{PE}(\tilde{\boldsymbol{X}}) = \sigma_\epsilon^2 + \mathrm{Bias}^2(\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}) + \mathrm{Var}(\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}})$$

- This decomposition is known as the **bias-variance tradeoff**:
  - As model becomes more complex (more terms included), model structure can be picked up
  - But coefficient estimates suffer from high variance as more terms are included in the model
- Introducing a little bias in our estimate for $\boldsymbol{\beta}$ might lead to a substantial decrease in variance, and hence to a substantial decrease in PE:
  - If the β_i's are unconstrained they can explode and hence are susceptible to very high variance
  - To control variance, we might regularize the coefficients, i.e., $\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|$

Bias-variance tradeoff

- Prediction error:

$$\mathrm{PE}(\tilde{\boldsymbol{X}}) = \mathrm{Bias}^2(\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}) + \mathrm{Var}(\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}) + \sigma_\epsilon^2$$

$$\mathrm{PE}(\tilde{\boldsymbol{X}}) = \mathbb{E}[(\hat{\boldsymbol{y}} - \boldsymbol{y})^2] + \sigma_\epsilon^2 = \mathbb{E}[(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}))^2] + \sigma_\epsilon^2$$

- Formula to compute variance:

$$\begin{aligned}
\mathrm{Var}(Y) &= \mathbb{E}[(Y - \mathbb{E}[Y])^2] \\
&= \mathbb{E}[Y^2 - 2Y\mathbb{E}[Y] + \mathbb{E}[Y]^2] \\
&= \mathbb{E}[Y^2] - 2\mathbb{E}[Y]\mathbb{E}[Y] + \mathbb{E}[Y]^2 \\
&= \mathbb{E}[Y^2] - \mathbb{E}[Y]^2
\end{aligned}$$

- Re-arrange formula that computes variance:

$$\mathbb{E}[Y^2] = (\mathbb{E}[Y])^2 + \mathrm{Var}(Y) = \underbrace{(\mathbb{E}[Y])^2}_{\mathrm{Bias}^2} + \underbrace{\mathbb{E}[(Y - \mathbb{E}[Y])^2]}_{\mathrm{Var}}$$

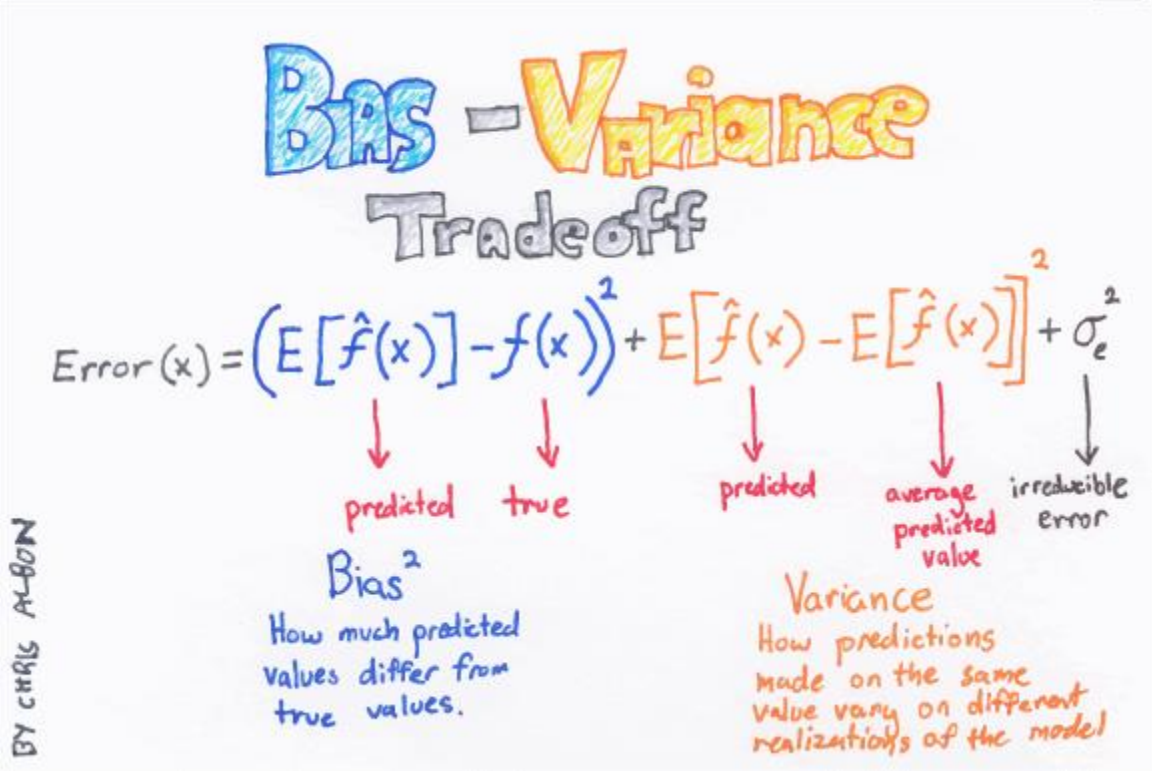
- Substitute $Y = \hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})$

Bias-variance tradeoff

- Prediction error:

$$\mathrm{PE}(\tilde{\boldsymbol{X}}) = \mathrm{Bias}^2(\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}) + \mathrm{Var}(\tilde{\boldsymbol{X}}\hat{\boldsymbol{\beta}}) + \sigma_\epsilon^2$$

$$\mathrm{PE}(\tilde{\boldsymbol{X}}) = \mathbb{E}[(\hat{\boldsymbol{y}} - \boldsymbol{y})^2] + \sigma_\epsilon^2 = \mathbb{E}[(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}))^2] + \sigma_\epsilon^2$$



Source: Machine Learning Flashcard by Chris Albon

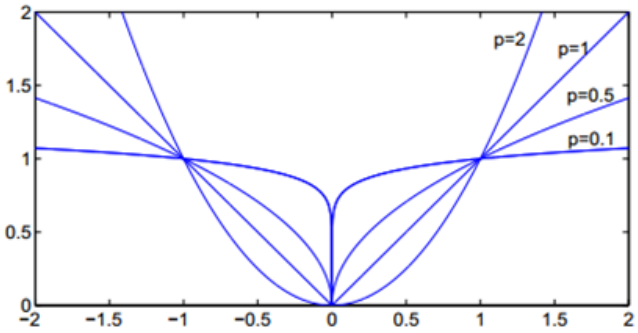
Regularized regression

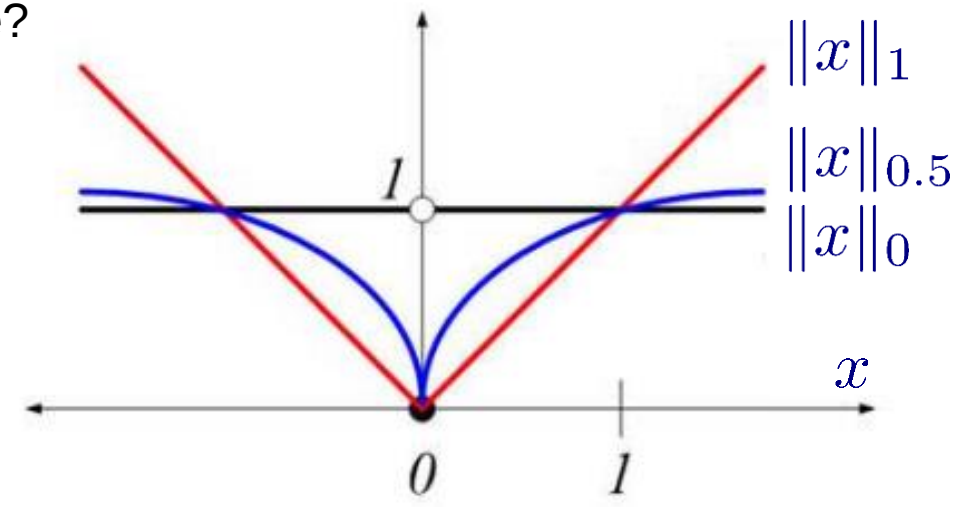
- **Regularized regression** is **multi-objective optimization problem**:

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \quad \Leftarrow \quad f_1(\boldsymbol{\beta})$$
$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{\beta}\| \quad \Leftarrow \quad f_2(\boldsymbol{\beta})$$

- **Two questions**:
  - How to solve this multi-objective optimization problem?
  - Which norm function $\|\cdot\|_p$ to choose?

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$





$$\text{card}(\boldsymbol{x}) = \sum_i x_i^0 = \|\boldsymbol{x}\|_0 \quad \text{with } 0^0 = 0$$

Multi-objective optimization

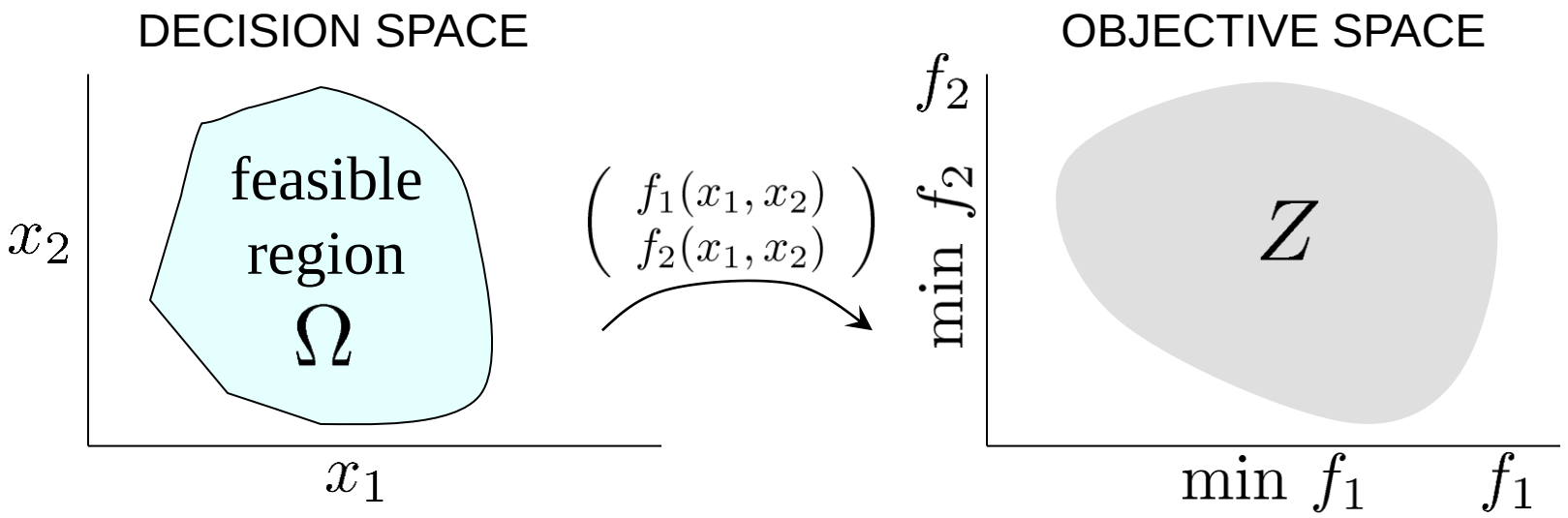
- Multi-objective optimization:

$$\begin{aligned} \min \quad & \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})\} \\ \text{s.t} \quad & \boldsymbol{x} \in \Omega \end{aligned}$$

$f_i : \mathbb{R}^n \to \mathbb{R}$ are (possibly) conflicting objectives and $\Omega \subseteq \mathbb{R}^n$ is feasible region

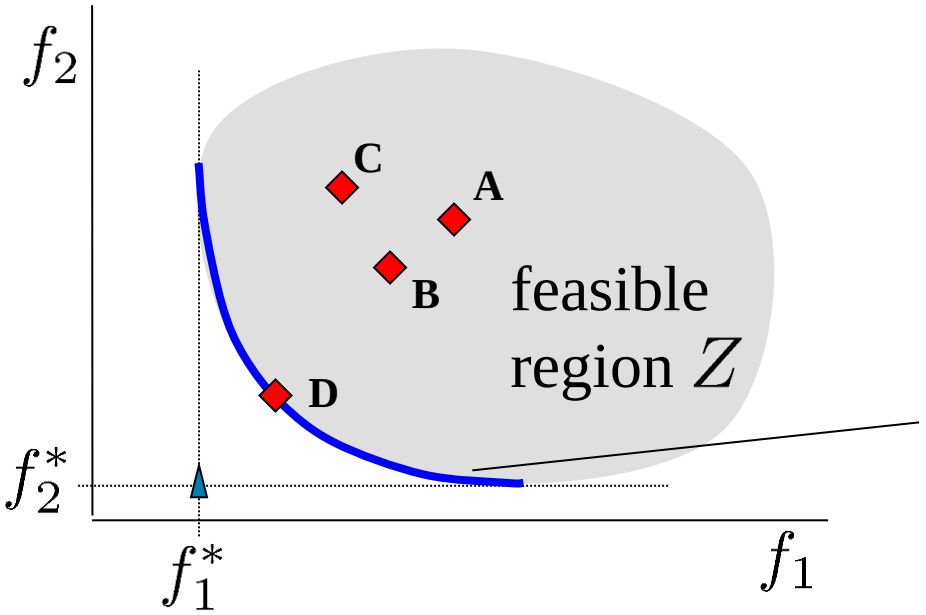
- Mapping feasible region into the objective space:

$$Z = \{\boldsymbol{z} \in \mathbb{R}^k \ : \ \boldsymbol{z} = ((f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T \ \forall \, \boldsymbol{x} \in \Omega)\}$$



Bi-objective example

- $\min\ f_1 = \text{bias}\,,\ \min\ f_2 = \text{variance}$:



Pareto frontier *or* ***efficient frontier***
(all non-dominated solutions)

- Pareto efficiency: solutions with characteristics like **D**, are called tradeoff, Pareto optimal or non-dominated

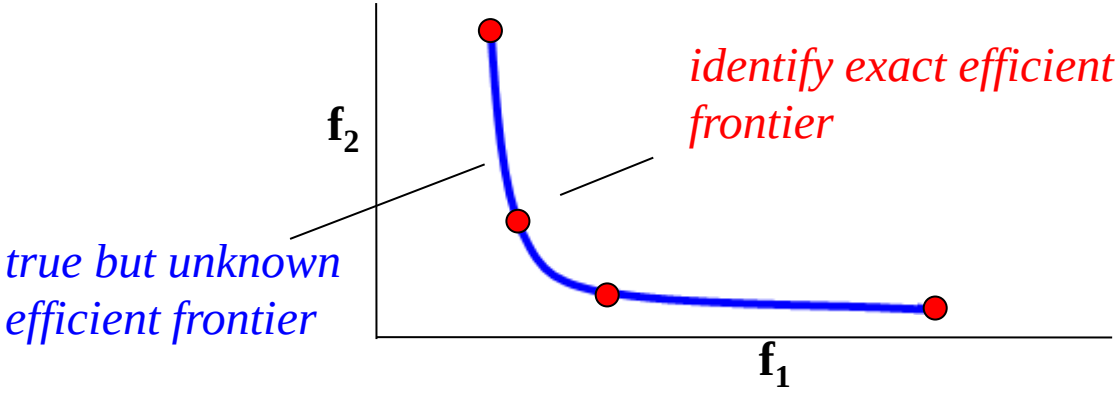
- **Multi-objective optimization goal**: find solution(s) on the efficient frontier according to the decision maker preferences

Computing efficient frontiers - possibilities

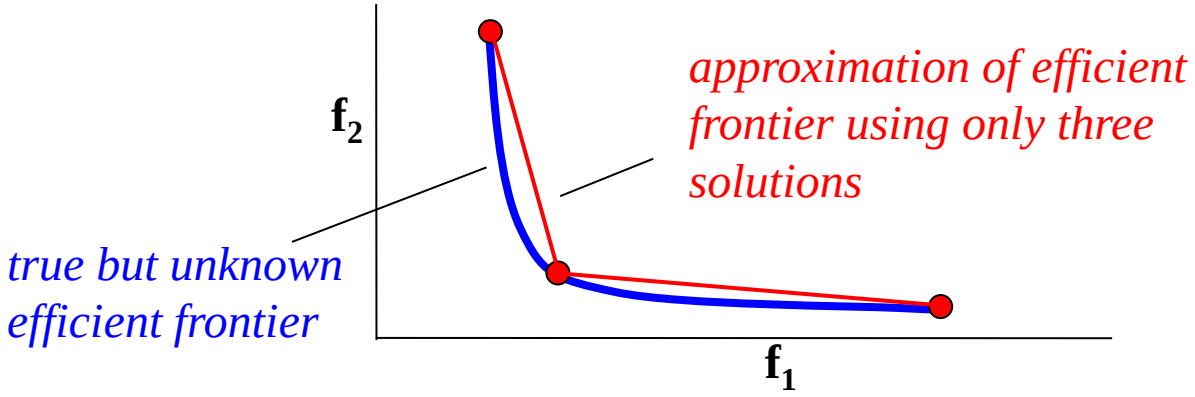
Multi-objective analysis involves **computing the efficient frontier**, evaluating it (if possible, out-of-sample) and selecting the final solution based on the decision maker preferences

Computing efficient frontiers:

- Ideal (often unrealistic) goal: compute exact frontier



- Typical (more realistic) goal: approximate the frontier



Solving multi-objective optimization problems

- **Convert multi-objective optimization problem to a series of single-objective optimization problems**

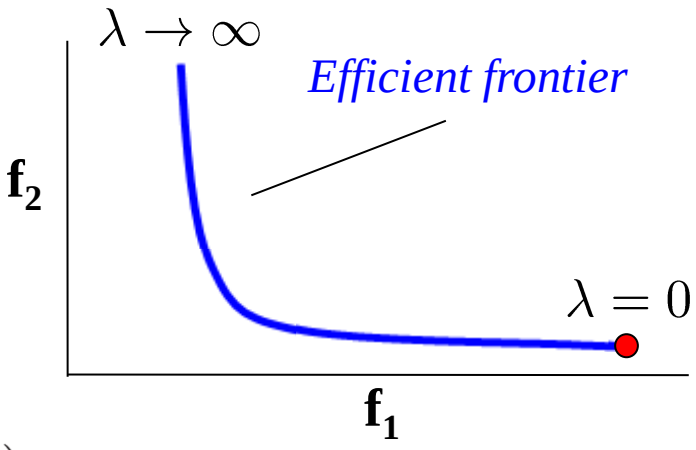
- Methods:
  - Weighting Method
  - ε-Constraint (Hierarchical) Method

Weighting method

- Assign weights to each objective
- Optimize the weighted sum of the objectives
- Multi-objective optimization with weighting method:

$$\begin{aligned} \min \quad & \omega_1 \cdot f_1(\boldsymbol{x}) + \omega_2 \cdot f_2(\boldsymbol{x}) \\ \text{s.t} \quad & \boldsymbol{x} \in \Omega \end{aligned}$$

f_i is convex function (linear, convex quadratic)
$\Omega \subseteq \mathbb{R}^n$ (convex)
$\omega_i \in \mathbb{R}$ is the weight of the i-th objective
$\omega_i \geq 0,\ i = 1, 2$ and $\omega_1 + \omega_2 = 1$



- Easier formulation:

$$\lambda = \frac{\omega_1}{\omega_2}$$

$$\begin{aligned} \min \quad & \lambda f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) \\ \text{s.t} \quad & \boldsymbol{x} \in \Omega \end{aligned}$$

ε - constrained method

- Optimize one objective
- Convert other objectives into constraints

- Multi-objective optimization with ε-constrained method:

First step

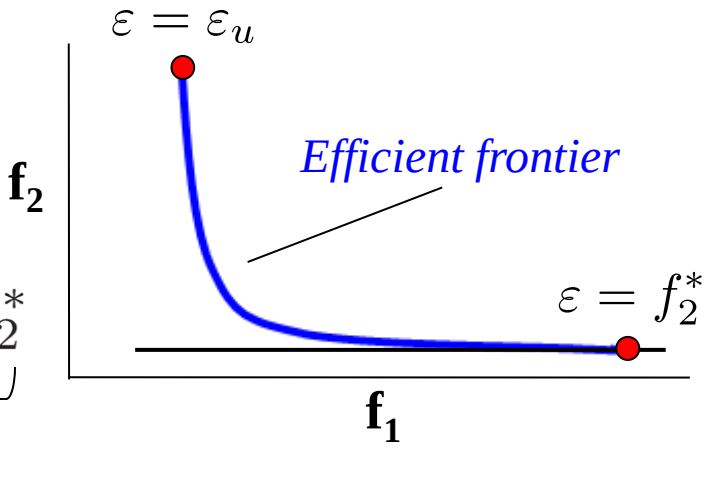
$$\begin{aligned} \min \quad & f_2(\boldsymbol{x}) \\ \text{s.t} \quad & \boldsymbol{x} \in \Omega \end{aligned}$$

$$\Downarrow$$

$$f_2^*$$

Second step

$$\begin{aligned} \min \quad & f_1(\boldsymbol{x}) \\ \text{s.t} \quad & \boldsymbol{x} \in \Omega \\ & f_2(\boldsymbol{x}) \leq \underbrace{(1+\epsilon) f_2^*}_{\varepsilon} \end{aligned}$$



Regularized regression

- **l_1 regularized regression and variable selection** (LASSO algorithm):

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$$

$$\Downarrow$$

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{\beta}\|_1 \leq \varepsilon \end{aligned}$$

- **l_2 regularized regression or Tikhonov regularization** (Ridge regression):

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

$$\Downarrow$$

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{\beta}\|_2^2 \leq \varepsilon \end{aligned}$$

Regularized regression

- ***l_1* regularized regression** (LASSO algorithm):

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$$

$$\Downarrow$$

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & (\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}) \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{n} |\beta_i| \leq \varepsilon \end{aligned}$$

- ***l_2* regularized regression** (Ridge regression):

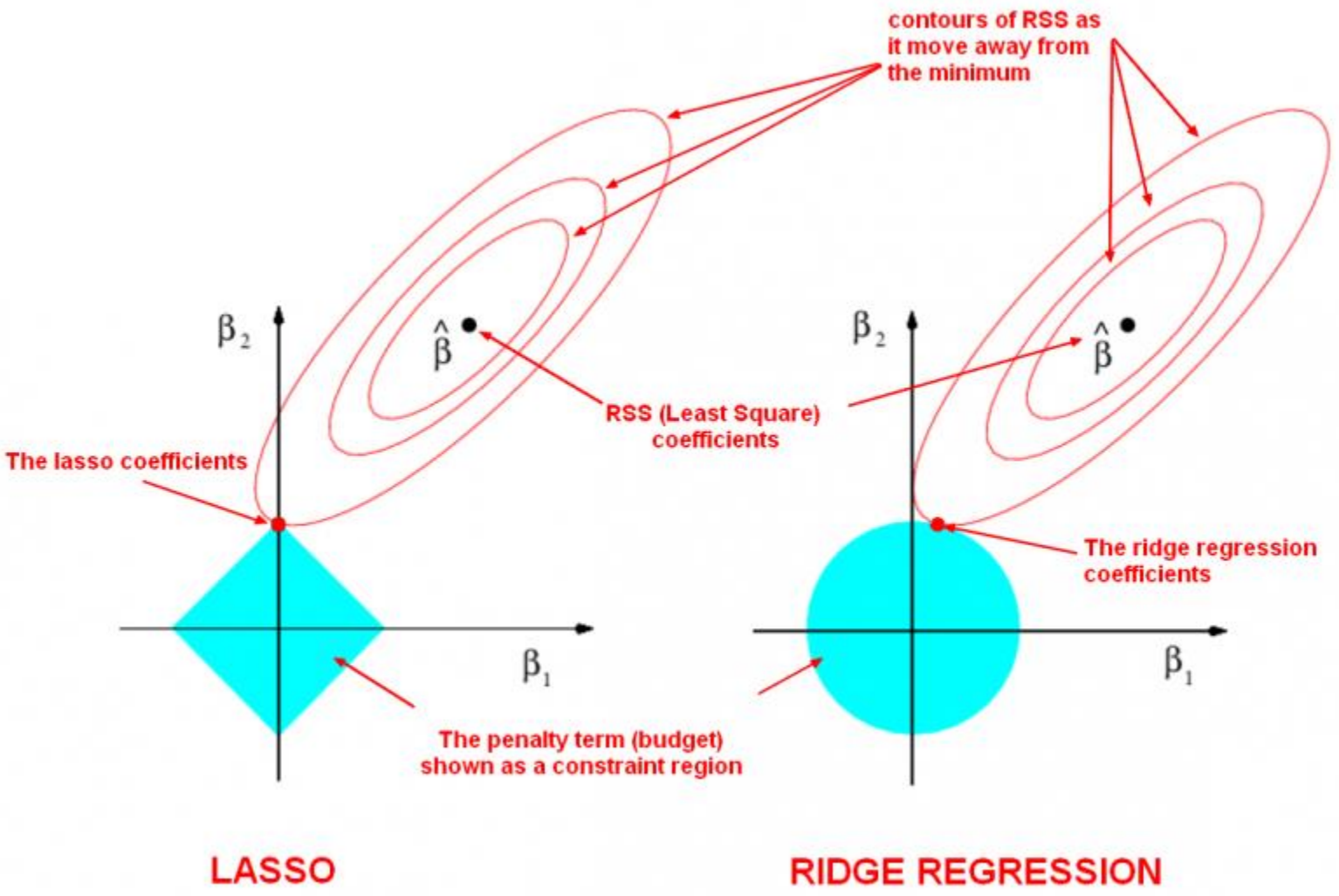
$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

$$\Downarrow$$

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & (\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}) \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{n} \beta_i^2 \leq \varepsilon \end{aligned}$$

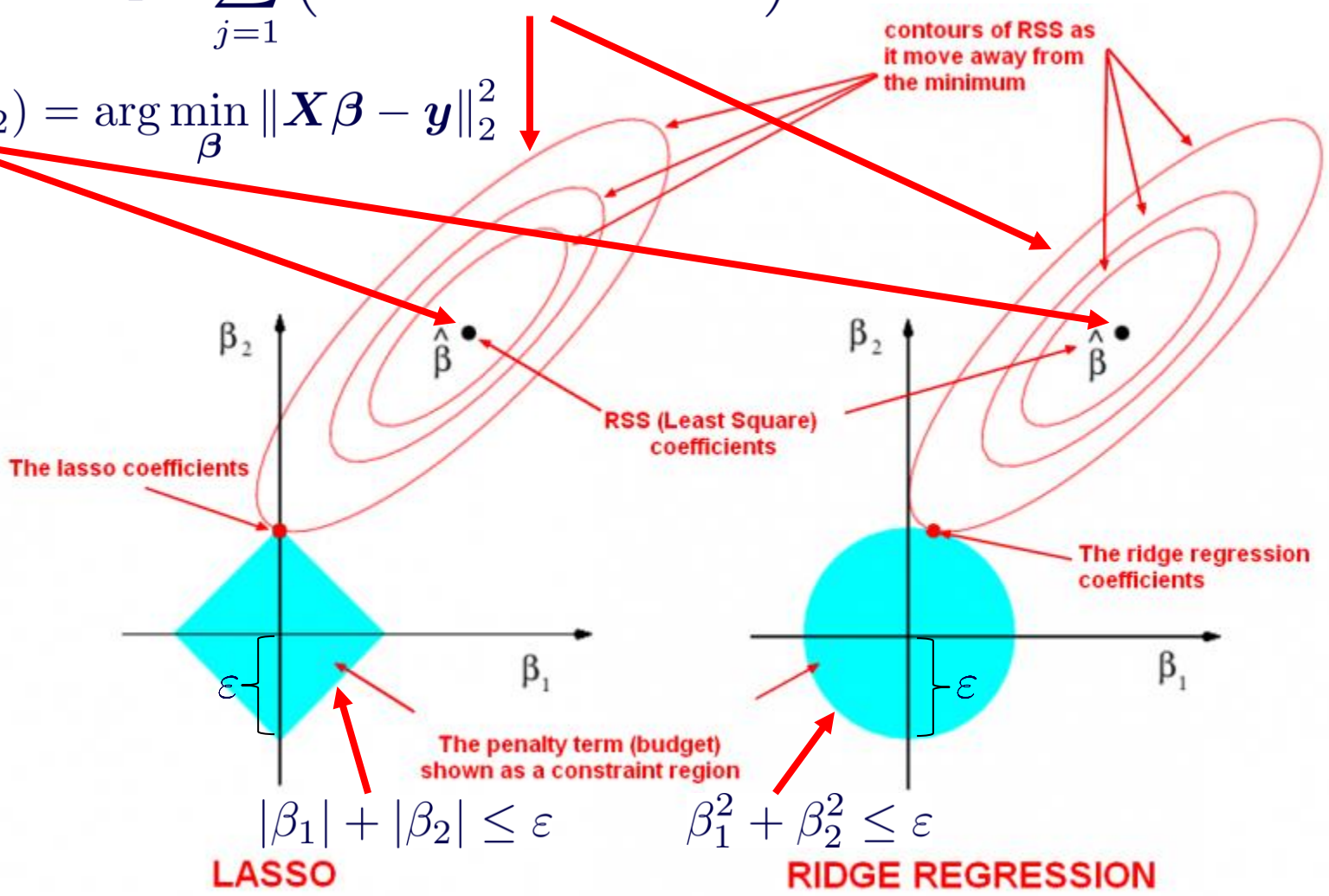
Regularized regression



Regularized regression

$$\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 = \sum_{j=1}^{m} \left(\beta_1 x_1^{(j)} + \beta_2 x_2^{(j)} - y^{(j)}\right)^2$$

$$(\hat{\beta}_1, \hat{\beta}_2) = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$$



Regularized regression

- ***l_2* regularized regression** (Ridge regression) has closed-form solution:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

- ***l_1* regularized regression** (LASSO algorithm) does not have a closed-form solution:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$$

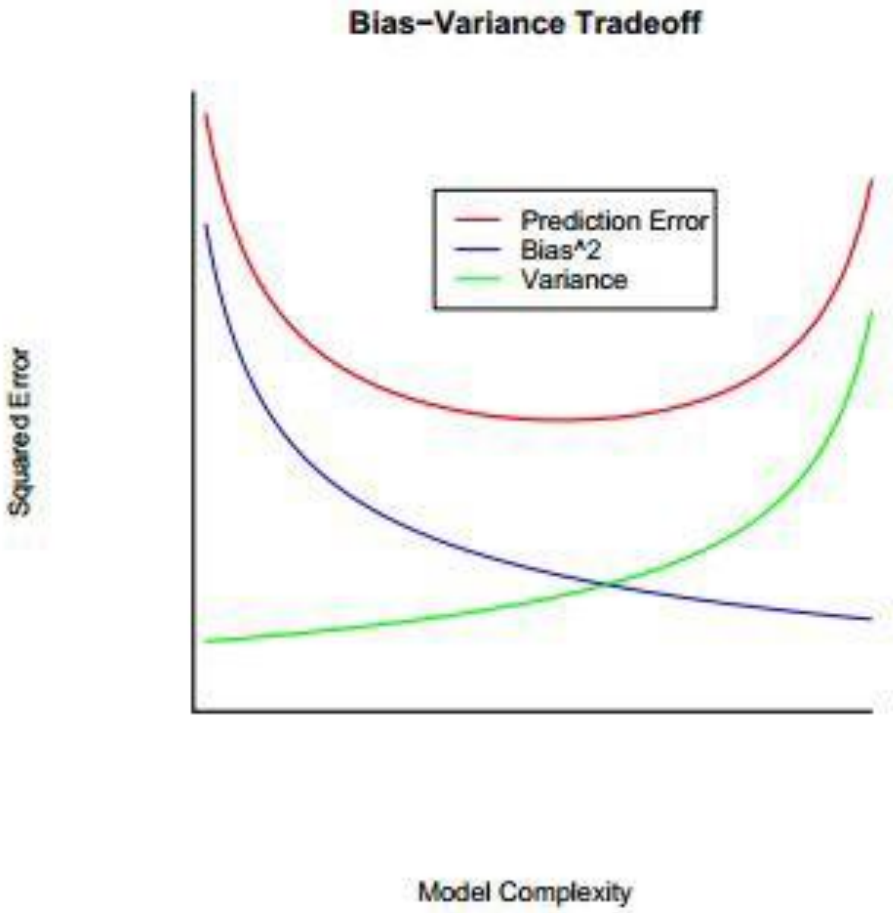
$$\min_{\boldsymbol{\beta}} \quad \boldsymbol{\beta}^T \left(\boldsymbol{X}^T\boldsymbol{X}\right)\boldsymbol{\beta} - 2\left(\boldsymbol{X}^T\boldsymbol{y}\right)^T\boldsymbol{\beta} + \boldsymbol{y}^T\boldsymbol{y}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \underbrace{|\beta_i|}_{\overline{\beta}_i + \underline{\beta}_i} \leq \varepsilon$$

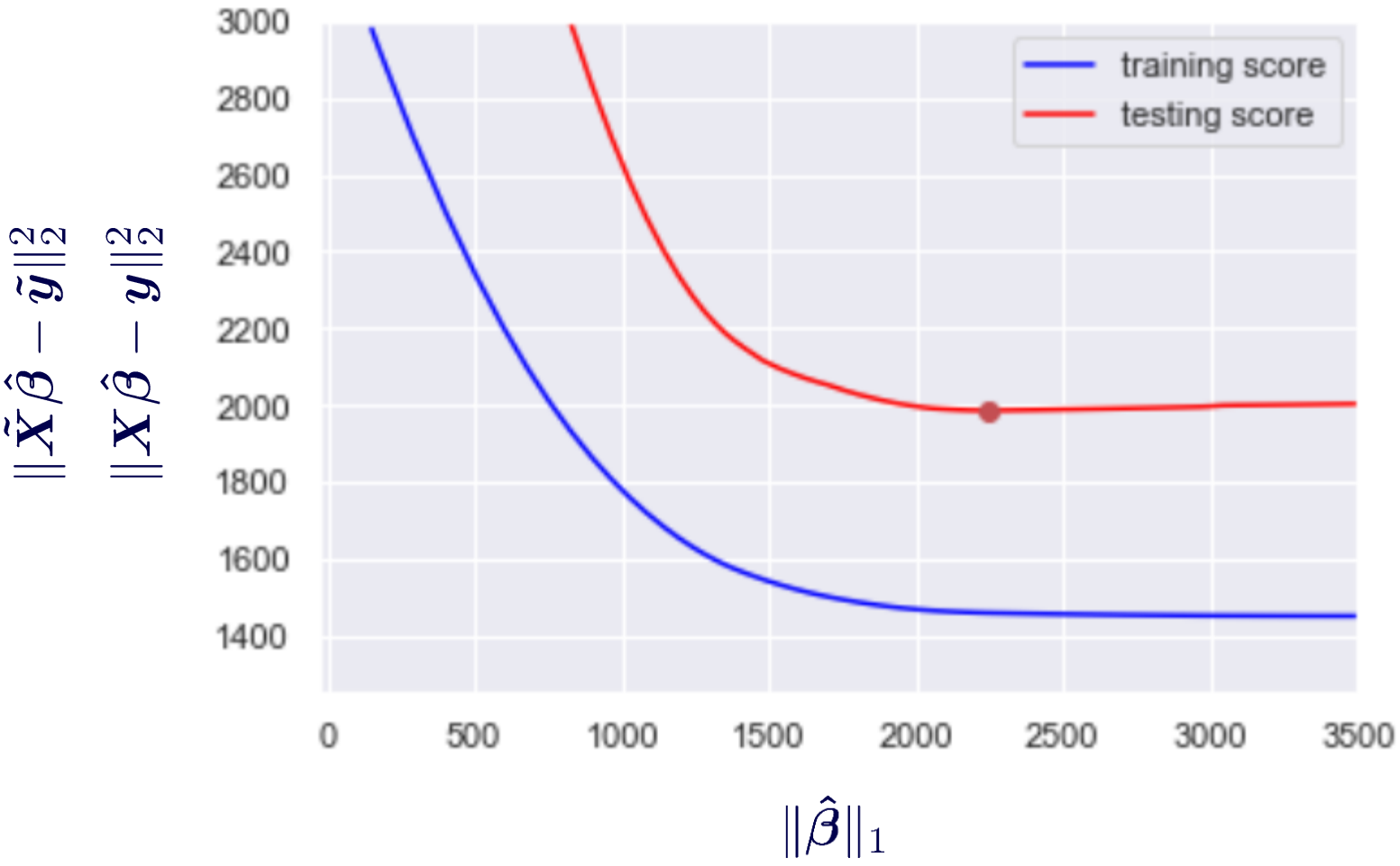
$$\begin{cases} |\beta_i| = \overline{\beta}_i + \underline{\beta}_i \\ \beta_i = \overline{\beta}_i - \underline{\beta}_i \\ \overline{\beta}_i \geq 0, \ \underline{\beta}_i \geq 0 \end{cases}$$

Optimization ensures that in an optimum $\overline{\beta}_i = 0$ or $\underline{\beta}_i = 0$

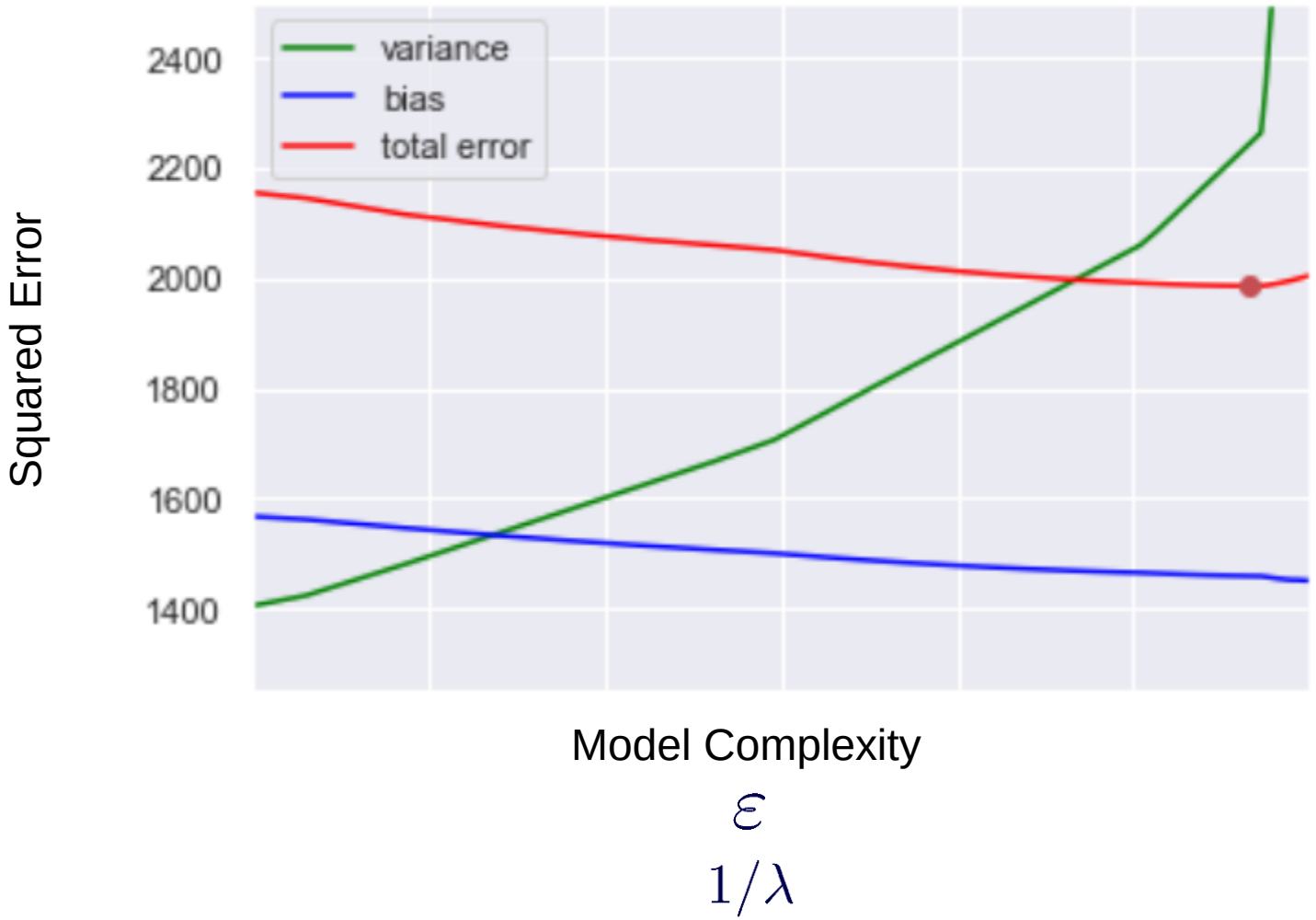
Regularized regression – bias-variance tradeoff



Regularized regression – bias-variance tradeoff



Regularized regression – bias-variance tradeoff



Regularized regression – hyperparameter tuning in `sklearn`

- ***l_1* regularized linear regression**:

$$\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{m} \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2}_{J(\boldsymbol{\theta})} + \alpha \|\boldsymbol{\theta}\|_1$$

MSE-loss → $J(\boldsymbol{\theta})$

hyperparameter → α

- ***l_1* regularized logistic regression**:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_1 + C \cdot J(\boldsymbol{\theta})$$

log-loss → $J(\boldsymbol{\theta})$

Regularized regression – hyperparameter tuning in `sklearn`

- ***l_1* regularized linear regression**:

$$\min_{\boldsymbol{\theta}} \ \frac{1}{m} \left\| \boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2 + \alpha \left\| \boldsymbol{\theta} \right\|_1$$

`sklearn` notation

$$\min_{\boldsymbol{\theta}} \ \left\| \boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2 + \underbrace{\alpha \cdot m}_{\lambda = \alpha \cdot m} \left\| \boldsymbol{\theta} \right\|_1$$

multiply objective function by positive number `m`

$$\min_{\boldsymbol{\beta}} \left\| \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y} \right\|_2^2 + \lambda \left\| \boldsymbol{\beta} \right\|_1$$

lecture notation

- ***l_1* regularized logistic regression**:

$$\min_{\boldsymbol{\theta}} \ \left\| \boldsymbol{\theta} \right\|_1 + C \cdot J(\boldsymbol{\theta})$$

`sklearn` notation

$$\min_{\boldsymbol{\theta}} \ J(\boldsymbol{\theta}) + \lambda \left\| \boldsymbol{\theta} \right\|_1$$

divide objective function by positive number `C` and denote $\lambda = 1/C$