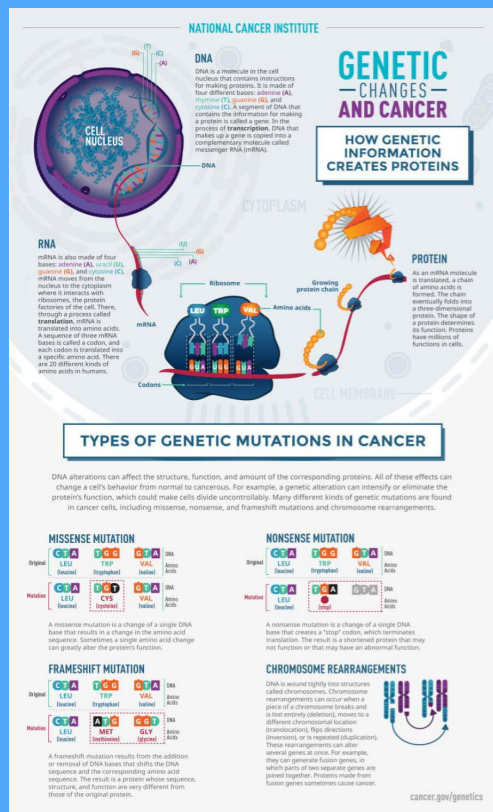


Introduction to Scientific Computation

Halil Bayraktar

Lecture 10 – Logistic Regression/Machine Learning



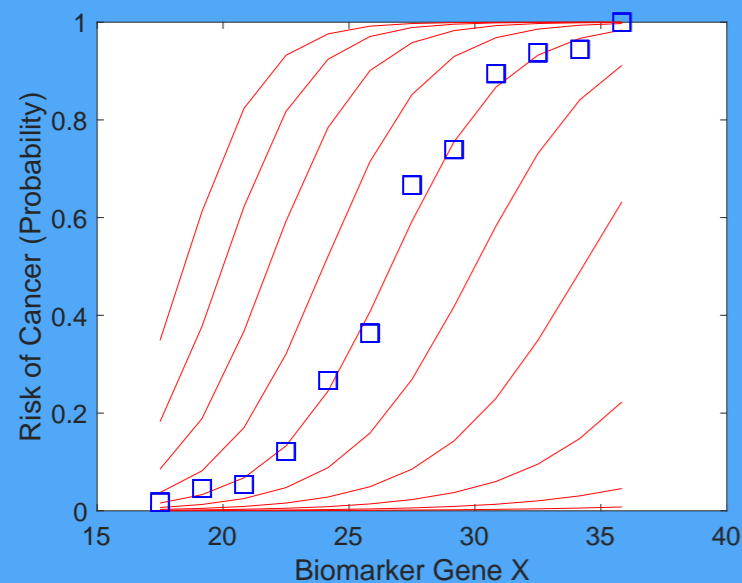
AlphaFold Protein Structure Database

Developed by Google DeepMind and EMBL-EBI

Search for protein, gene, UniProt accession or organism or sequence search

Examples: MENFQKVEKIGEGTYGV... Free fatty acid receptor 2 At1g58602 Q5VSL9 E. coli

[See search help](#) [Go to online course](#)



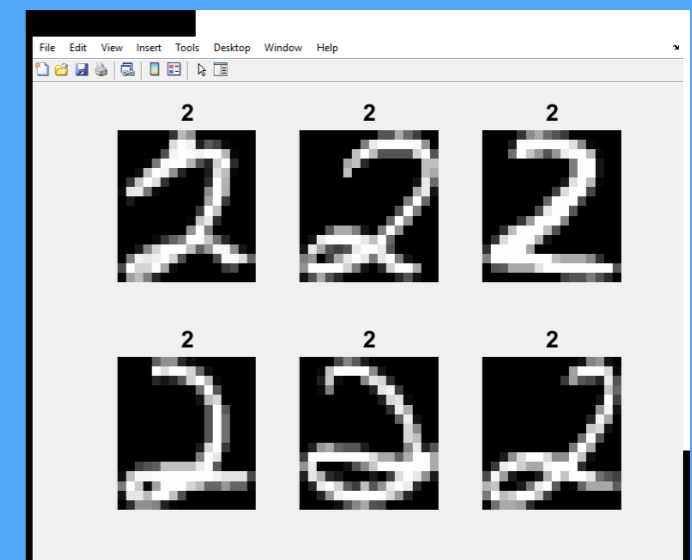
File Edit View Insert Tools Desktop Window Help

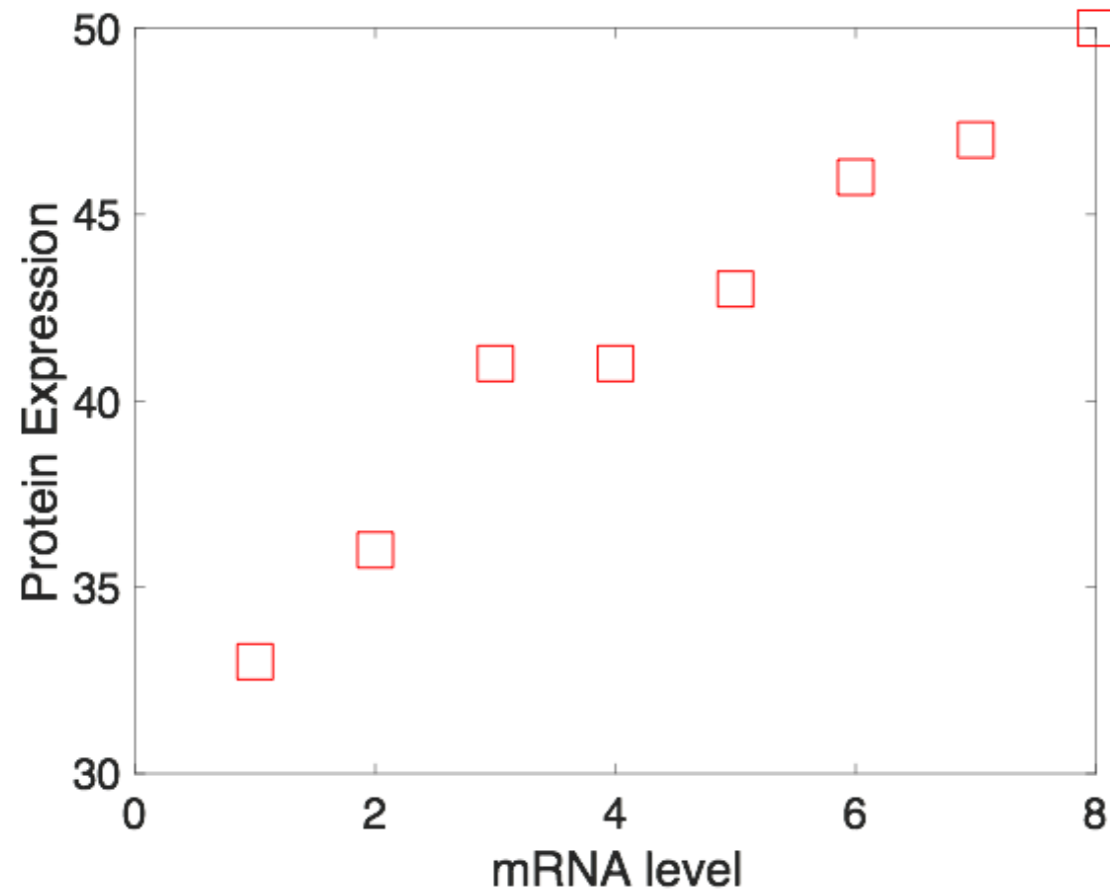
Confusion Matrix: Ensemble of Classification Trees

1	541	2				5		2	
2		544		3		3			
3	2		524	1	3	1	7	4	5
4	1		8	525		6	2	2	3
5			3		538		2		1
6	1	1		12	2	530	2		2
7	1	5	3		3		538		
8		1			4			538	2
9	1	4	8	6	2	6	2		506
10		1	1	1	6			6	3
	1	2	3	4	5	6	7	8	9
	1	2	3	4	5	6	7	8	9

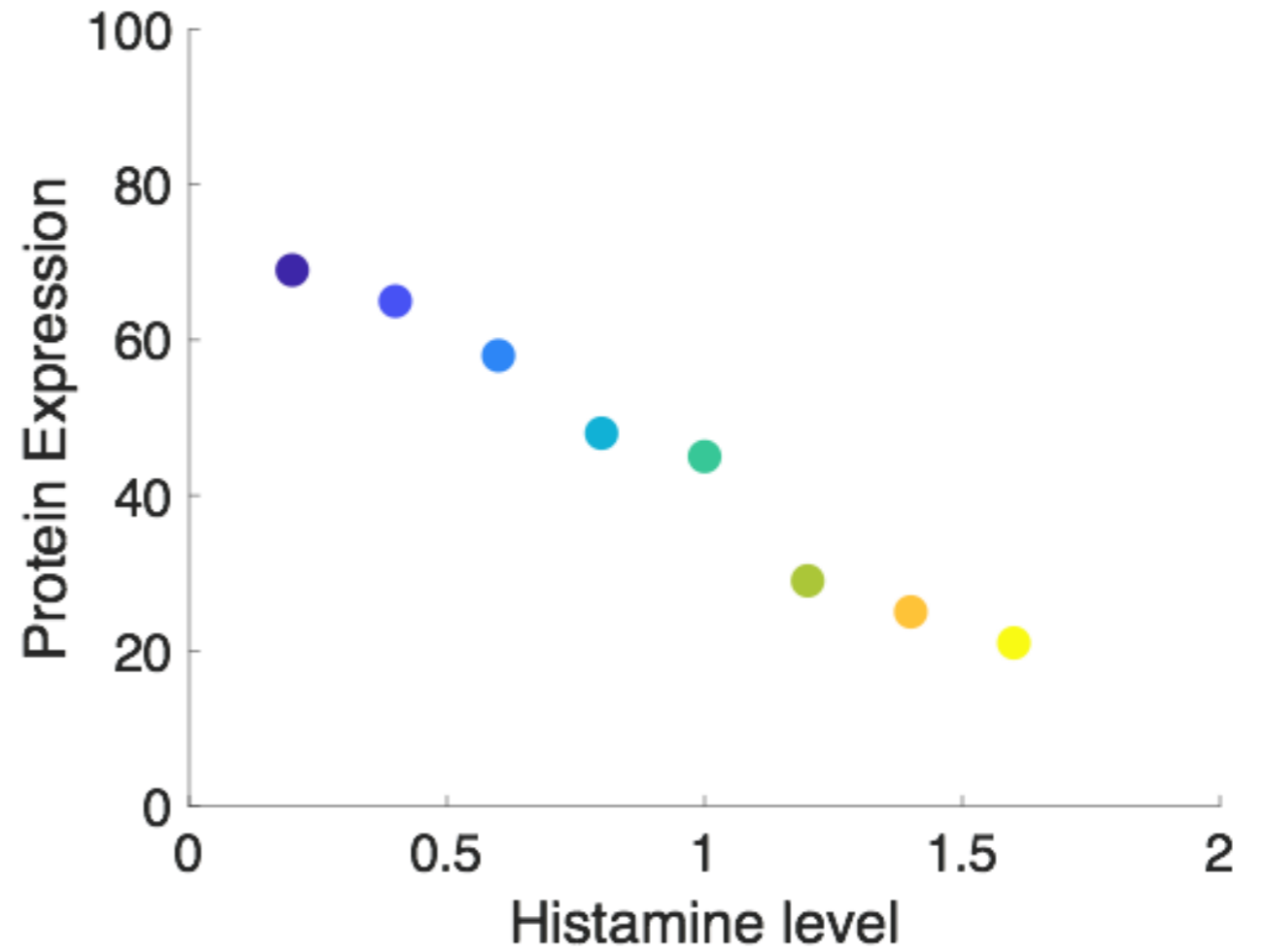
True Class

Predicted Class





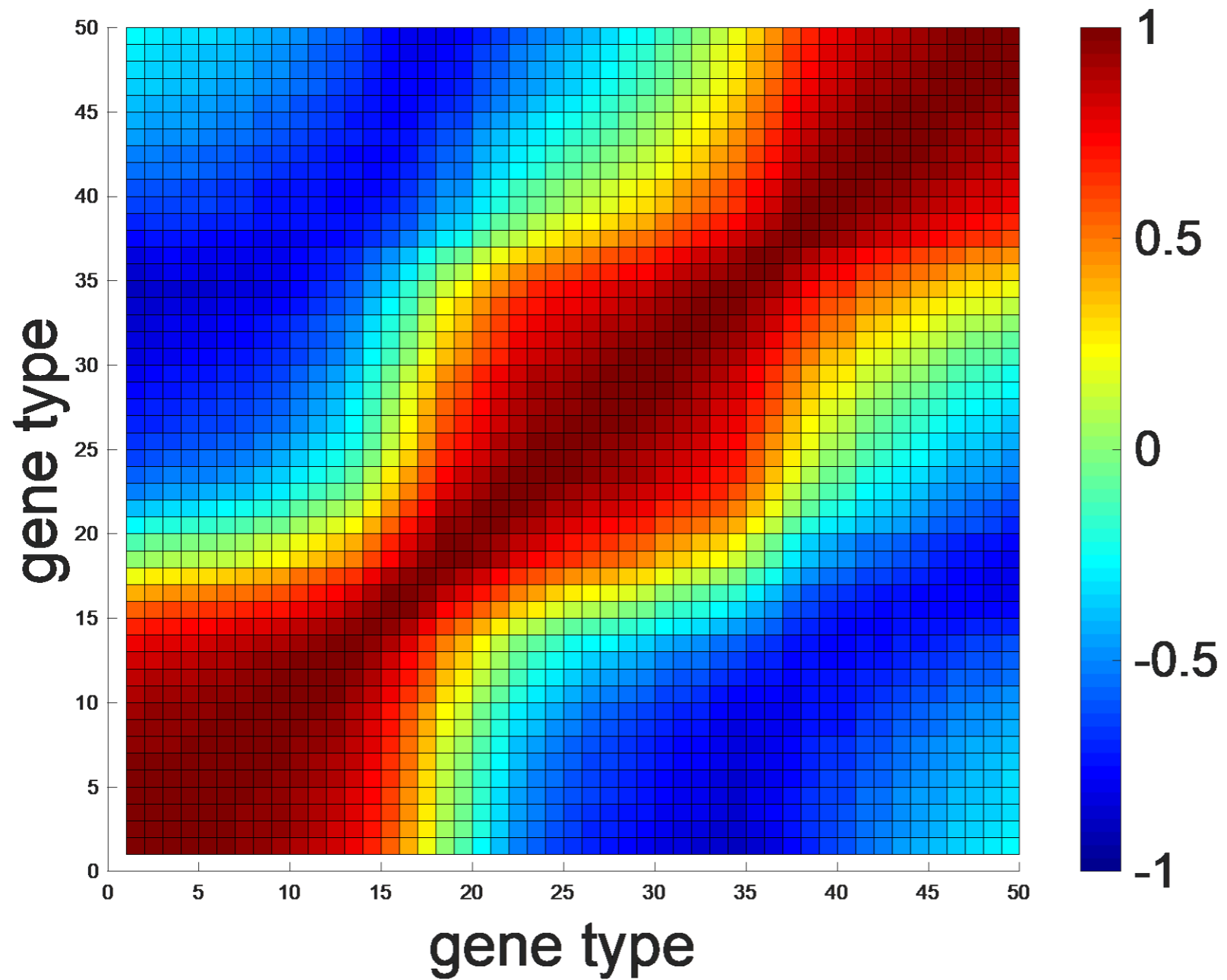
$r=+0.9538$



$r=-0.987$

Correlation coefficient always lies between -1 to +1

Correlation of difference genes



MULTIVARIATE REGRESSION

- y - dependent variable or also called response variable
- $x_1, x_2, x_3 \dots, x_n$ are called independent variables

or explanatory variables.

- X values can either quantitative or categorical.

$$Y = \text{constant (a)} + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \dots + \beta_k x_n$$

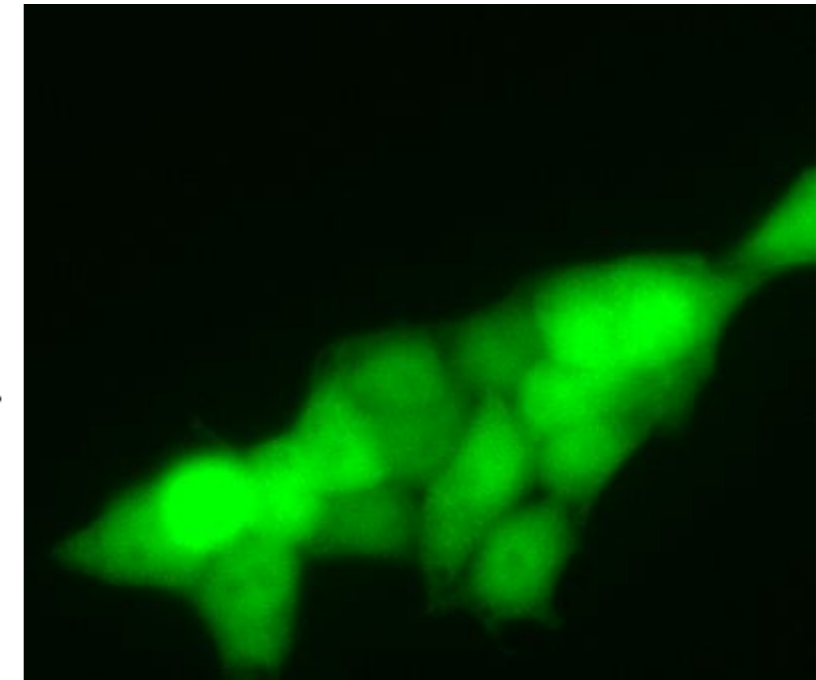
Dependence of cell growth to expression of geneX, geneY and geneZ

Linear regression model:

$$\underline{y} \sim 1 + x1 + x2 + x3$$

Estimated Coefficients:

	<u>Estimate</u>	<u>SE</u>	<u>tStat</u>	<u>pValue</u>
(Intercept)	47.153	26.499	1.7794	0.078342
x1	0.28602	0.069679	4.1048	8.4971e-05
x2	-0.0033967	0.0047938	-0.70856	0.48031
x3	-0.3098	0.071258	-4.3476	3.4254e-05



Number of observations: 100, Error degrees of freedom: 96

Root Mean Squared Error: 1.74

R-squared: 0.994, Adjusted R-Squared 0.993

F-statistic vs. constant model: 4.95e+03, p-value = 4.52e-105

>>

$$\text{Cell growth} = 47 + 0.28\text{geneX} - 0.003\text{geneY} - 0.30\text{geneZ}$$

Example 2.

Dependence of fuel consumption to car features (weight, horse power, model year etc.)



Linear regression model:
 $y \sim 1 + x1 + x2 + x3$

Estimated Coefficients:

	<u>Estimate</u>	<u>SE</u>	<u>tStat</u>	<u>pValue</u>
(Intercept)	47.977	3.8785	12.37	4.8957e-21
x1	<u>-6.5416</u>	1.1274	-5.8023	<u>9.8742e-08</u>
x2	-0.042943	0.024313	-1.7663	<u>0.08078</u>
x3	-0.011583	0.19333	-0.059913	<u>0.95236</u>

Number of observations: 93, Error degrees of freedom: 89
Root Mean Squared Error: 4.09
R-squared: 0.752, Adjusted R-Squared 0.744
F-statistic vs. constant model: 90, p-value = 7.38e-27
>> |



A one-unit difference in the rating of weight corresponds to a 6.5 point difference in fuel consumption.

Logistic Regression

If a response variable such as yes/no or success/failure response variables., we cannot use linear regression models where it assumes a normal distribution.

Think about a cancer patient diagnosis whether a patient either have a cancer or not a cancer

One type of model that can be used is called **logistic regression**. We think in terms of a binomial model for the two possible values of the response variable and use one or more explanatory variables to explain the probability of success.

$$P(Y=1|\beta) = \frac{\exp(b(1)+b(2)x)}{1+\exp(b(1)+b(2)x)}$$

x= binary or cont

y= binary

b(1) and b(2) are coefficients

Odds Ratio

Odds ratio is the ratio of the proportions for binary outcomes.

If \hat{p} is the proportion for one outcome, then $1 - \hat{p}$ is the proportion for the second outcome:

$$\text{odds} = \frac{\hat{p}}{1 - \hat{p}}$$

$$P(\text{pass exam}) = 0.8$$

$$\text{odds}(\text{pass}) = 4$$

$$P(\text{fail exam}) = 0.2$$

Odd ration of being cancer after seeing biomarker changes

$$P(\text{cancer})=0.83$$

$$\text{Odds ration (cancer)}=5$$

$$P(\text{not cancer})=0.17$$

the odds for having cancer is 5 times

the odds for not having cancer after rna sequencing

if y response variable is discrete

پند
سوی

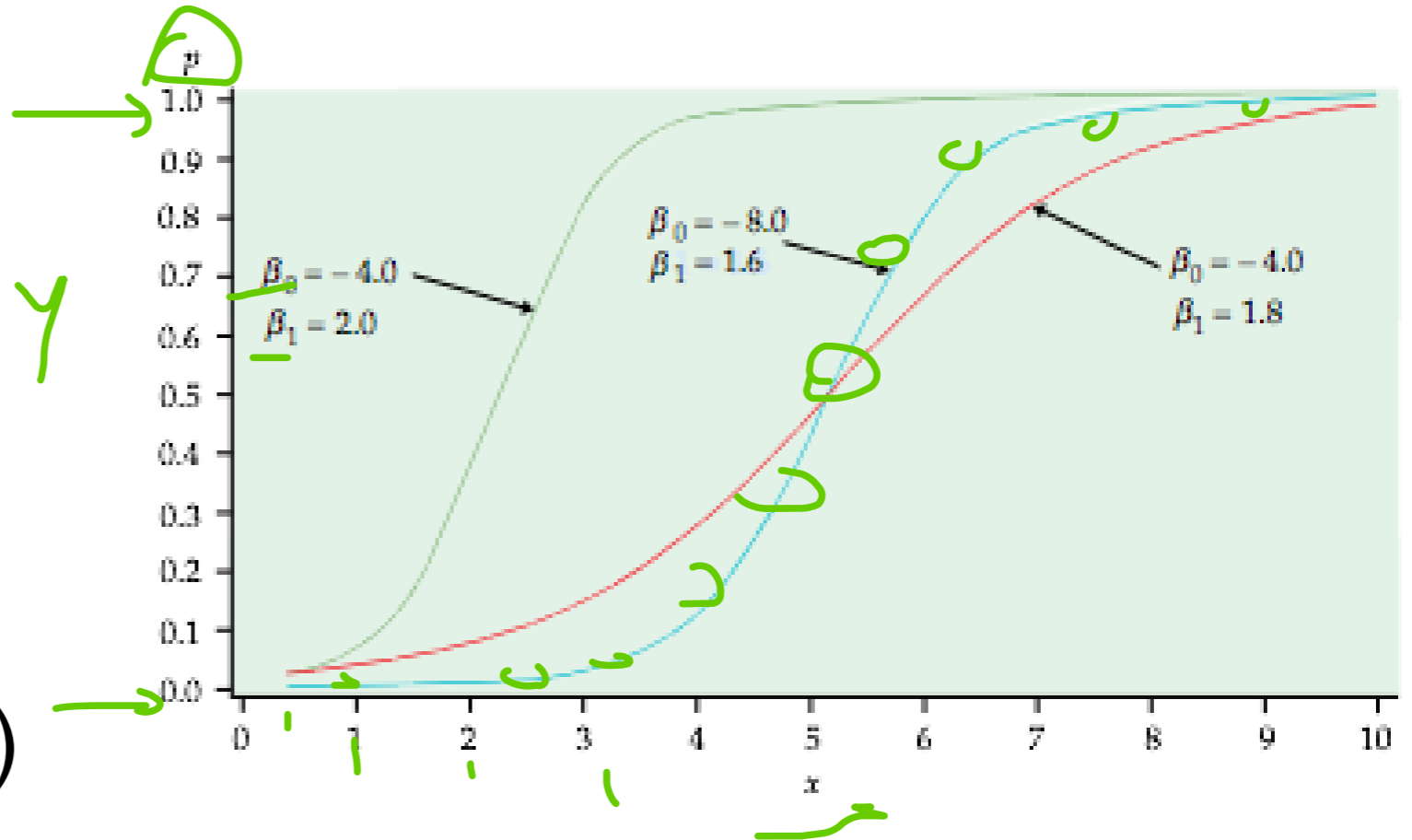
یافته‌ها

$$Y = P(Y=0) + P(Y=1)$$

Logistic function

it can be defined as

$$f(x) = \frac{\exp(x)}{1 + \exp(x)}$$



$f(x)$ or y values always falls in range between 0 and 1

$f(x)$ is probability values,

Logistic regression

Statistical model for logistic regression

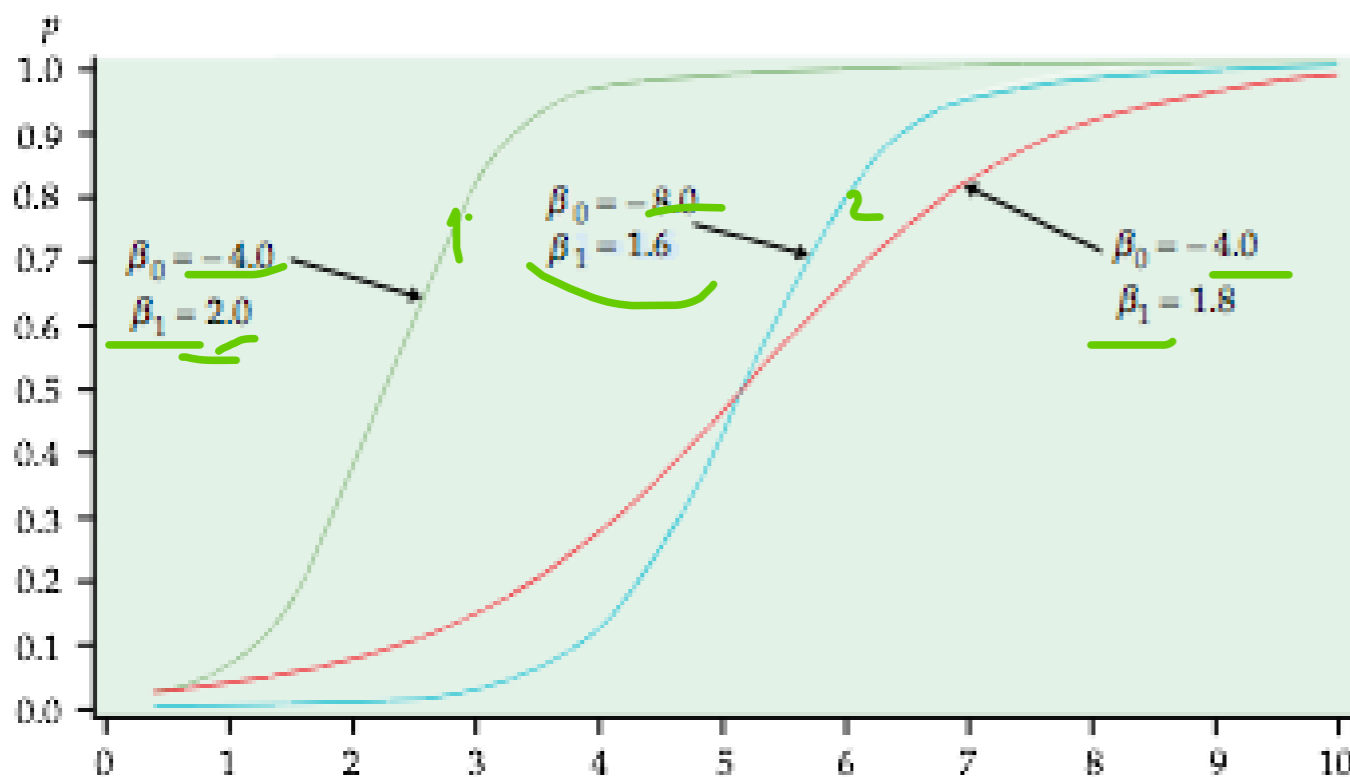
$$f(x) = \frac{\exp(x)}{1 + \exp(x)} \longleftrightarrow$$
$$p = \frac{\exp(x)}{1 + \exp(x)}$$

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x$$

Reverse, divide by exp(x) and rearrange we got

p is the probability value

beta values are coefficients of the logistic model



Derivation of log function

$$f(x) = \frac{\exp(x)}{1 + \exp(x)}$$

Reverse it

$$\frac{1}{p} = 1 + \frac{\exp(x)}{\exp(x)}$$

Divide by $\exp(x)$

$$\frac{1}{p} = \frac{1}{\exp(x)} + 1$$

Rearrange formular

$$\frac{1}{p} - 1 = \frac{1}{\exp(x)} + 1 \longrightarrow 1 - \frac{p}{p} = \frac{1}{\exp(x)}$$

$$\longrightarrow \frac{p}{p-1} = \exp(x)$$

Finally, take log of both sides, we got

$$\log\left(\frac{p}{p-1}\right) = b(1) + b(2)x$$

Example

We will classify the cells as normal (0) and cancerous 1
if gene expression > 65

Given that we have three variables

Exposure to radiation

Acidity

Reactive molecules

Logistic regression and cancer risk estimation

This example involves an experiment to help model the various gene expression levels that links to cancer occurrence. The data include observations of gene expression, patients tested, and number cancers.

NATIONAL CANCER INSTITUTE

GENETIC CHANGES AND CANCER

HOW GENETIC INFORMATION CREATES PROTEINS

DNA
DNA is a molecule in the cell nucleus that contains instructions for making proteins. It is made of four different bases: adenine (A), thymine (T), guanine (G), and cytosine (C). A segment of DNA that contains the information for making a protein is called a gene. In the process of **transcription**, DNA that makes up a gene is copied into a complementary molecule called messenger RNA (mRNA).

RNA
mRNA is also made of four bases: adenine (A), uracil (U), guanine (G), and cytosine (C). mRNA moves from the nucleus to the cytoplasm where it interacts with ribosomes, the protein factories of the cell. There, through a process called **translation**, mRNA is translated into amino acids. A sequence of three mRNA bases is called a codon, and each codon is translated into a specific amino acid. There are 20 different kinds of amino acids in humans.

PROTEIN
As an mRNA molecule is translated, a chain of amino acids is formed. The chain eventually folds into a three-dimensional protein. The shape of a protein determines its function. Proteins have millions of functions in cells.

TYPES OF GENETIC MUTATIONS IN CANCER

DNA alterations can affect the structure, function, and amount of the corresponding proteins. All of these effects can change a cell's behavior from normal to cancerous. For example, a genetic alteration can intensify or eliminate the protein's function, which could make cells divide uncontrollably. Many different kinds of genetic mutations are found in cancer cells, including missense, nonsense, and frameshift mutations and chromosome rearrangements.

MISSENSE MUTATION

Original	CTA LEU (leucine)	TGG TRP (tryptophan)	GTA VAL (valine)	DNA Amino Acids
Mutation	CTA LEU (leucine)	TGT CYS (cysteine)	GTA VAL (valine)	DNA Amino Acids

A missense mutation is a change of a single DNA base that results in a change in the amino acid sequence. Sometimes a single amino acid change can greatly alter the protein's function.

NONSENSE MUTATION

Original	CTA LEU (leucine)	TGG TRP (tryptophan)	GTA VAL (valine)	DNA Amino Acids
Mutation	CTA LEU (leucine)	TGA [stop]	GTA VAL (valine)	DNA Amino Acids

A nonsense mutation is a change of a single DNA base that creates a "stop" codon, which terminates translation. The result is a shortened protein that may not function or that may have an abnormal function.

FRAMESHIFT MUTATION

Original	CTA LEU (leucine)	TGG TRP (tryptophan)	GTA VAL (valine)	DNA Amino Acids
Mutation	CTA LEU (leucine)	ATG MET (methionine)	GGT GLY (glycine)	DNA Amino Acids

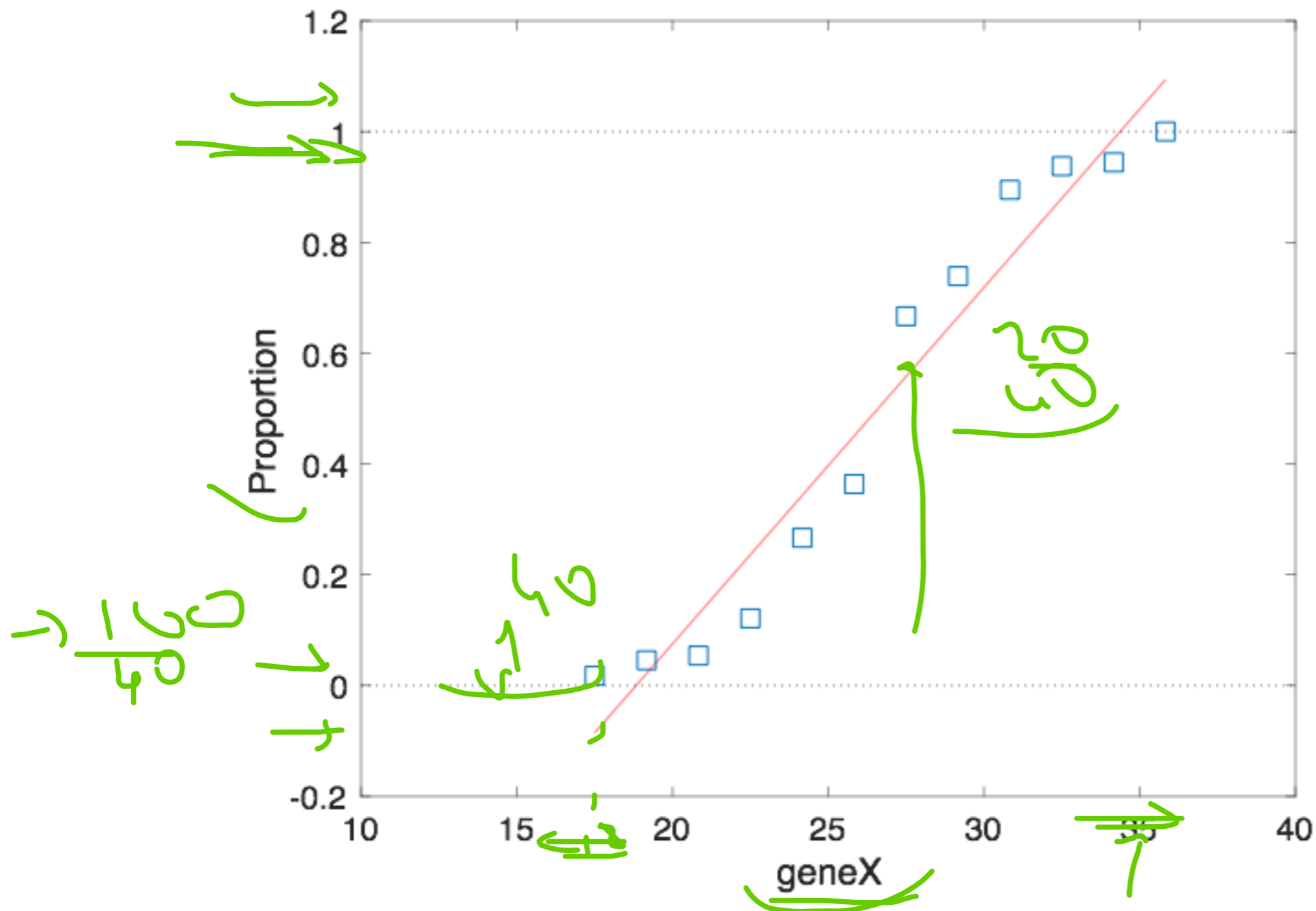
A frameshift mutation results from the addition or removal of DNA bases that shifts the DNA sequence and the corresponding amino acid sequence. The result is a protein whose sequence, structure, and function are very different from those of the original protein.

CHROMOSOME REARRANGEMENTS

DNA is wound tightly into structures called chromosomes. Chromosome rearrangements can occur when a piece of a chromosome breaks and is lost entirely (deletion), moves to a different chromosomal location (translocation), flips directions (inversion), or is repeated (duplication). These rearrangements can alter several genes at once. For example, they can generate fusion genes, in which parts of two separate genes are joined together. Proteins made from fusion genes sometimes cause cancer.

cancer.gov/genetics

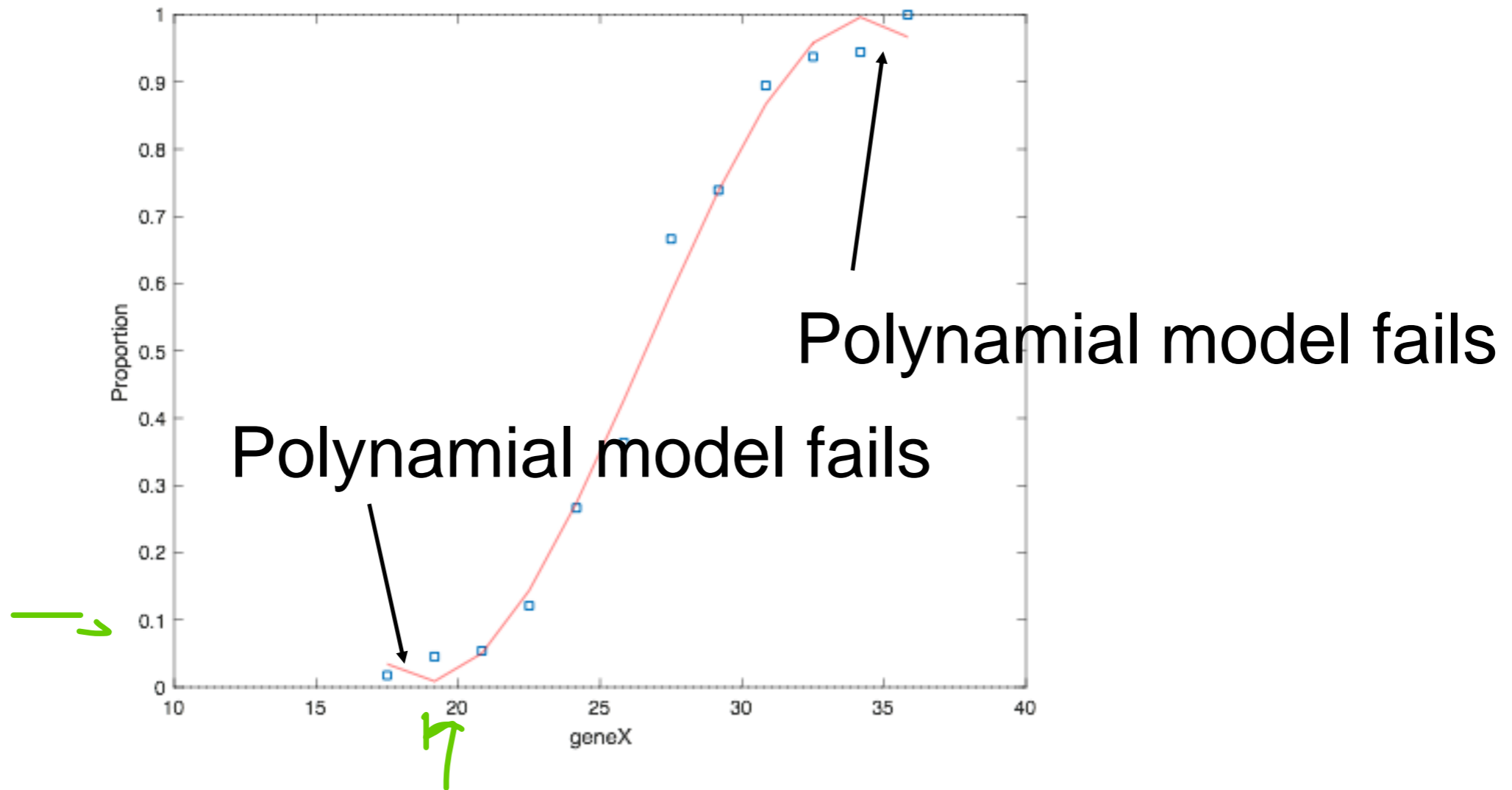
Lets fit with a linear model



However there are few problems if we use linear fit:

- 1) The fit line predicts proportions less than 0 and greater than 1 when proportion geneX level is at very high and low levels
- 2) Proportions are not normally distributed. This contradicts for fitting a simple linear regression model. It requires normal distribution.

Lets fit with a polynomial model

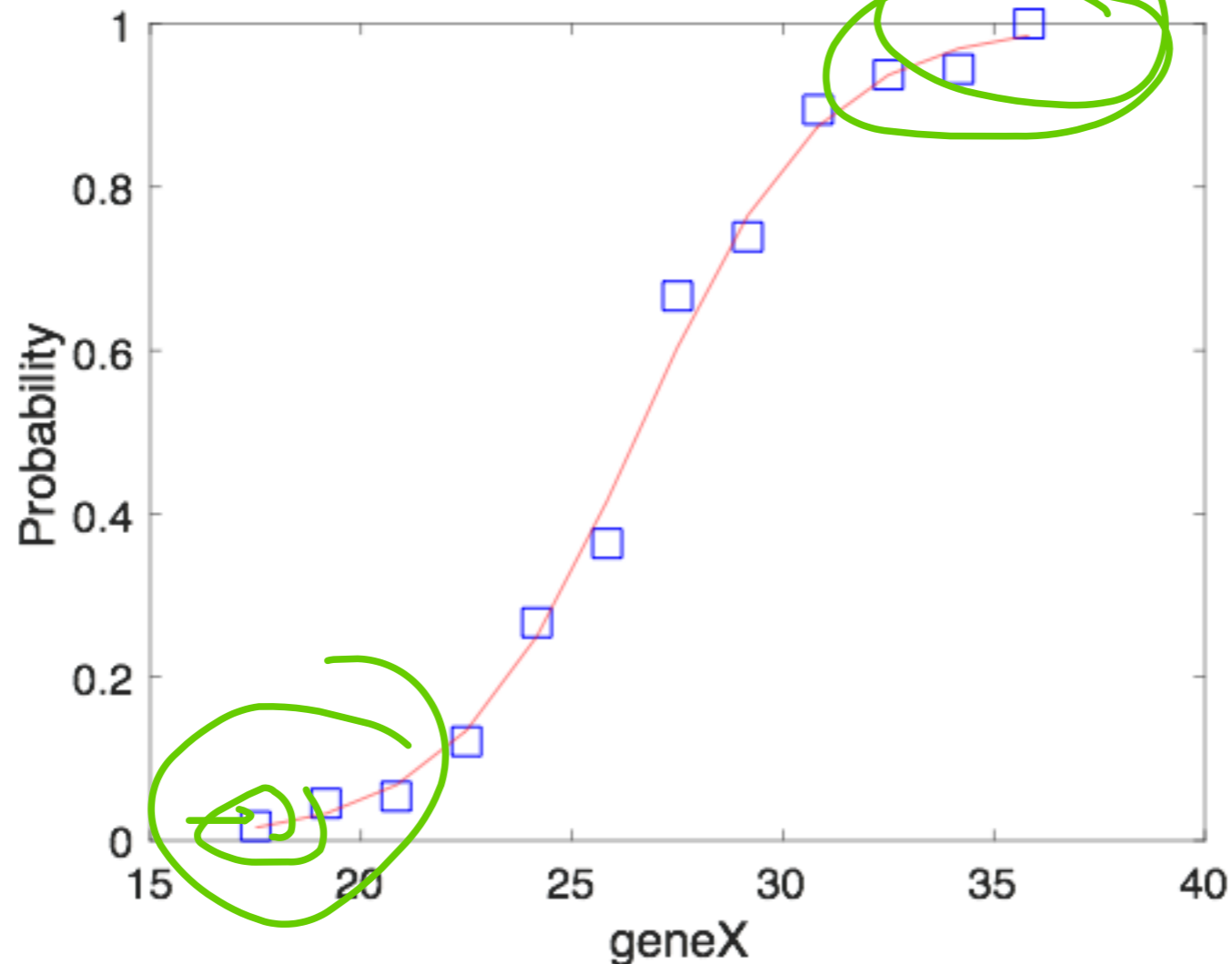


- 1) The fit line predicts proportions goes high and low values when proportion geneX level is at very high and low levels

Solutions: Logistic regression

Logistic regression is the best model if response variable is binomial. Because it uses a fitting method that is appropriate for the binomial distribution.

Predicted proportions/probability values are present in the range from 0 to 1.



In matlab we use glmfit function to fit our data to a logistic model. This function returns coefficient estimates for a linear regression of the responses Y ($f(x)$) on the independent variable X .

In Matlab,

```
%logistic regression
```

```
[logitCoef, dev, stats] = glmfit(geneX, [cancer  
tested], 'binomial', 'logit');
```

```

geneX = [2180 2450 2640 2730 3100 3120 3320 3610 3800
% The number of patients tested at each levels (intervals)
tested = [57 44 37 33 30 22 21 23 19 16 18 21]';
% The number of cancer patients at each test
cancer = [1 2 2 4 8 8 14 17 17 15 17 21]';

```

```
%logistic regression
```

```

[logitCoef,dev,stats] = glmfit(geneX,[cancer tested],'binomial','logit');
logitFit = glmval(logitCoef,geneX,'logit');

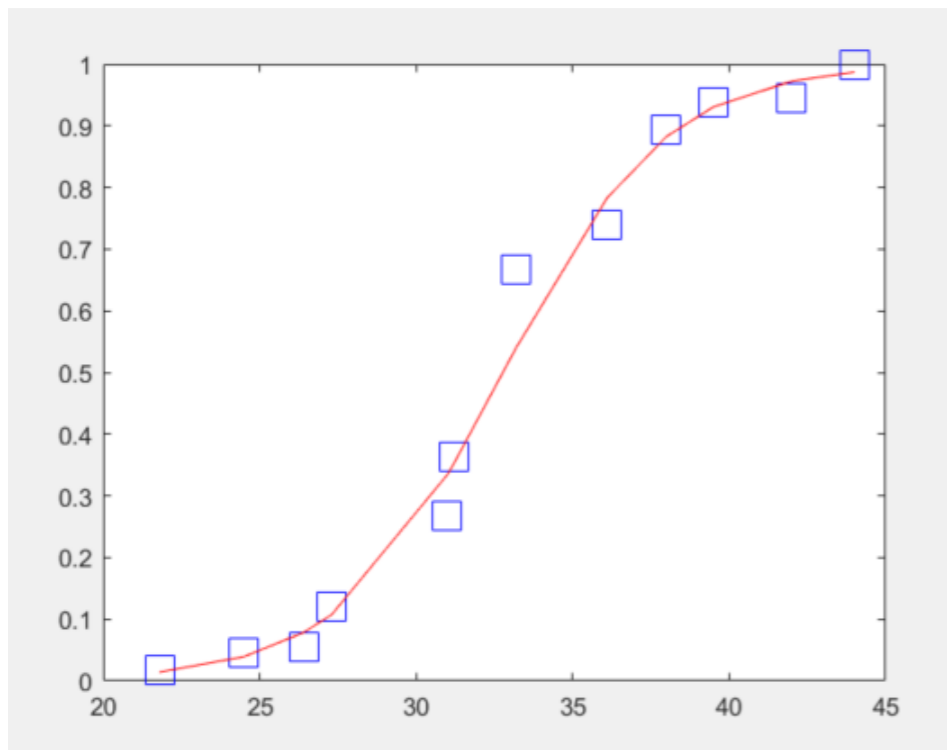
```

```
figure(3)
```

```
plot(geneX,proportion,'bs', geneX,logitFit,'r-','markersize',16);
```

	1	2	3	4	5
1	-12.6748				
2	0.3867				
3					
4					

Glmval is uses to compute the predicted values for the model



	1	2	3
1	0.0141		
2	0.0391		
3	0.0782		
4	0.1073		
5	0.3345		
6	0.3519		
7	0.5406		
8	0.7831		
9	0.8827		
10	0.9308		
11	0.9725		
12	0.9871		
13			
14			

glmfint: Logistic model coefficients

stats x logitFit x dev x logitCoef x

2x1 double

	1	2	3	4	5
1	-12.6748				
2	0.3867				
3					
4					

stats x logitFit x dev x logitCoef x

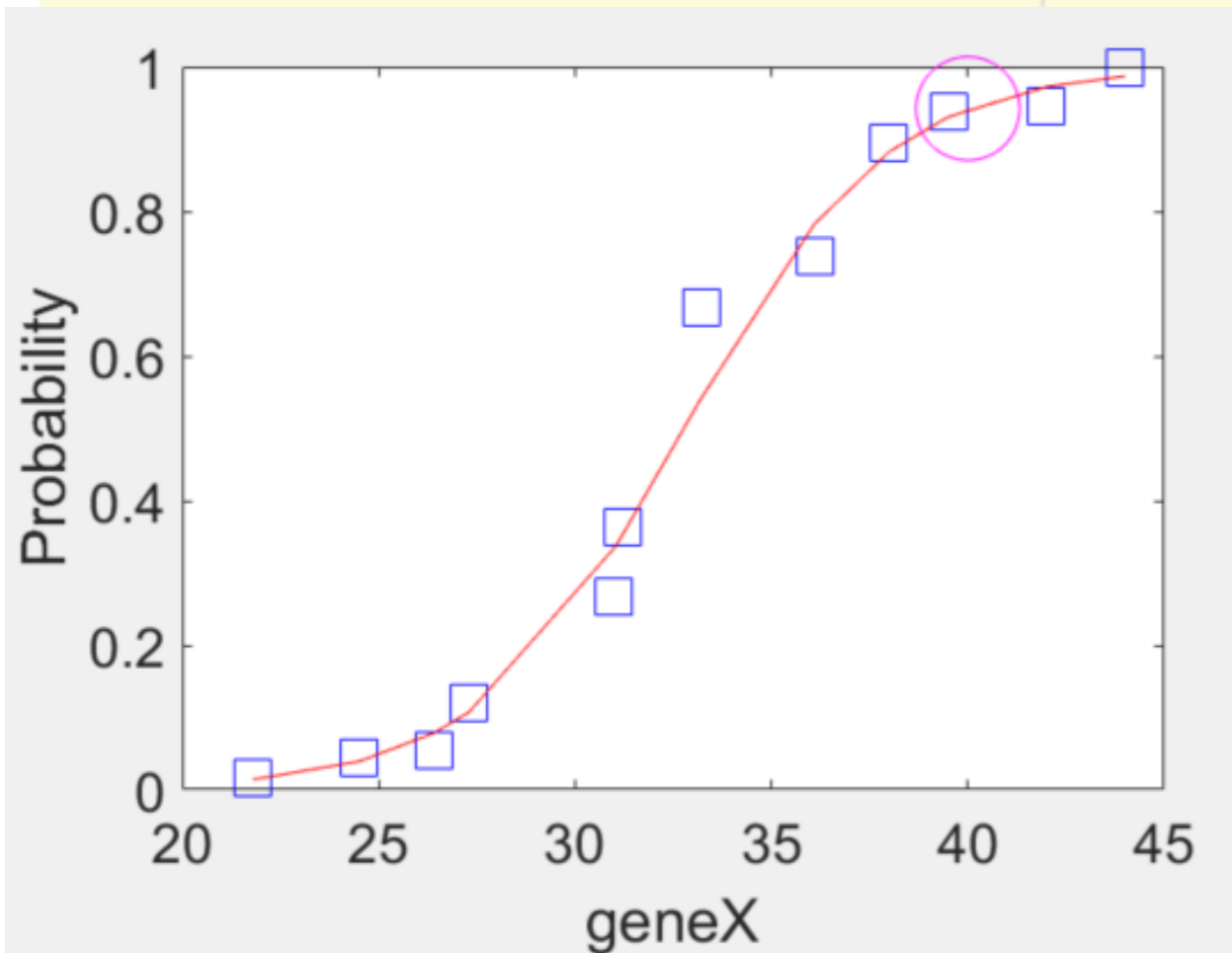
1x1 struct with 15 fields

Field ^	Value
beta	[-12.6748;0.3867]
dfe	10
sfit	0.5951
s	1
<input checked="" type="checkbox"/> estdisp	0
covb	[1.6374,-0.0508;-0.0508,0....
se	[1.2796;0.0400]
coeffcorr	[1,-0.9907;-0.9907,1]
t	[-9.9053;9.6573]
p	[3.9472e-23;4.5767e-22]
resid	12x1 double
residp	12x1 double
residd	12x1 double
resida	12x1 double
wts	12x1 double

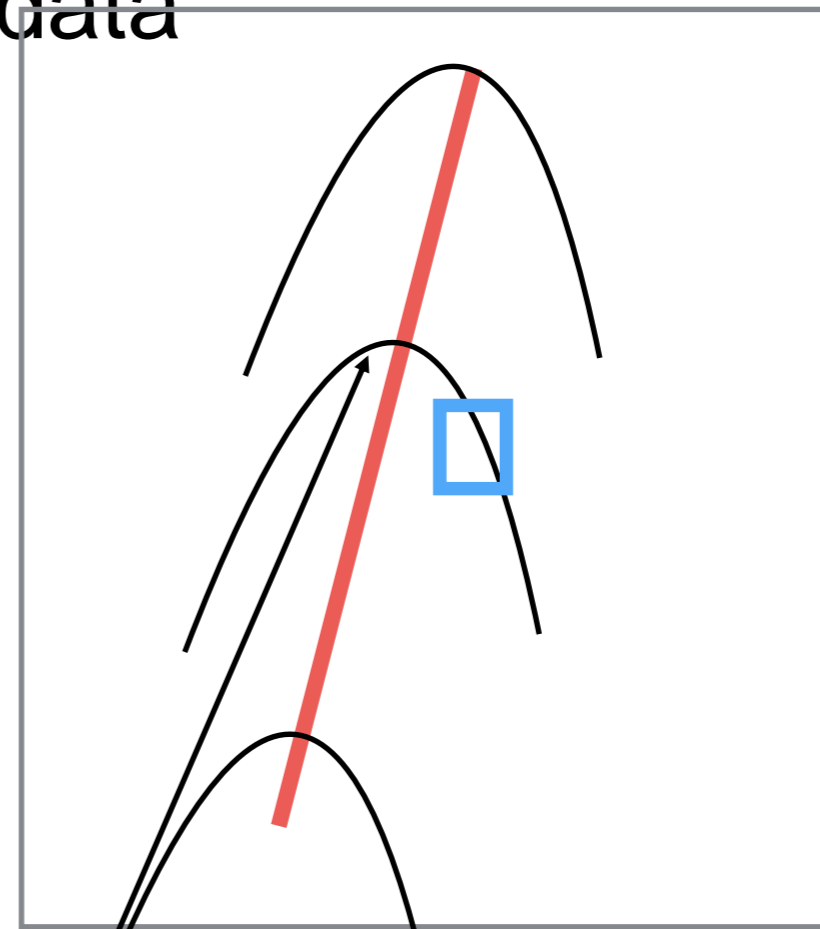
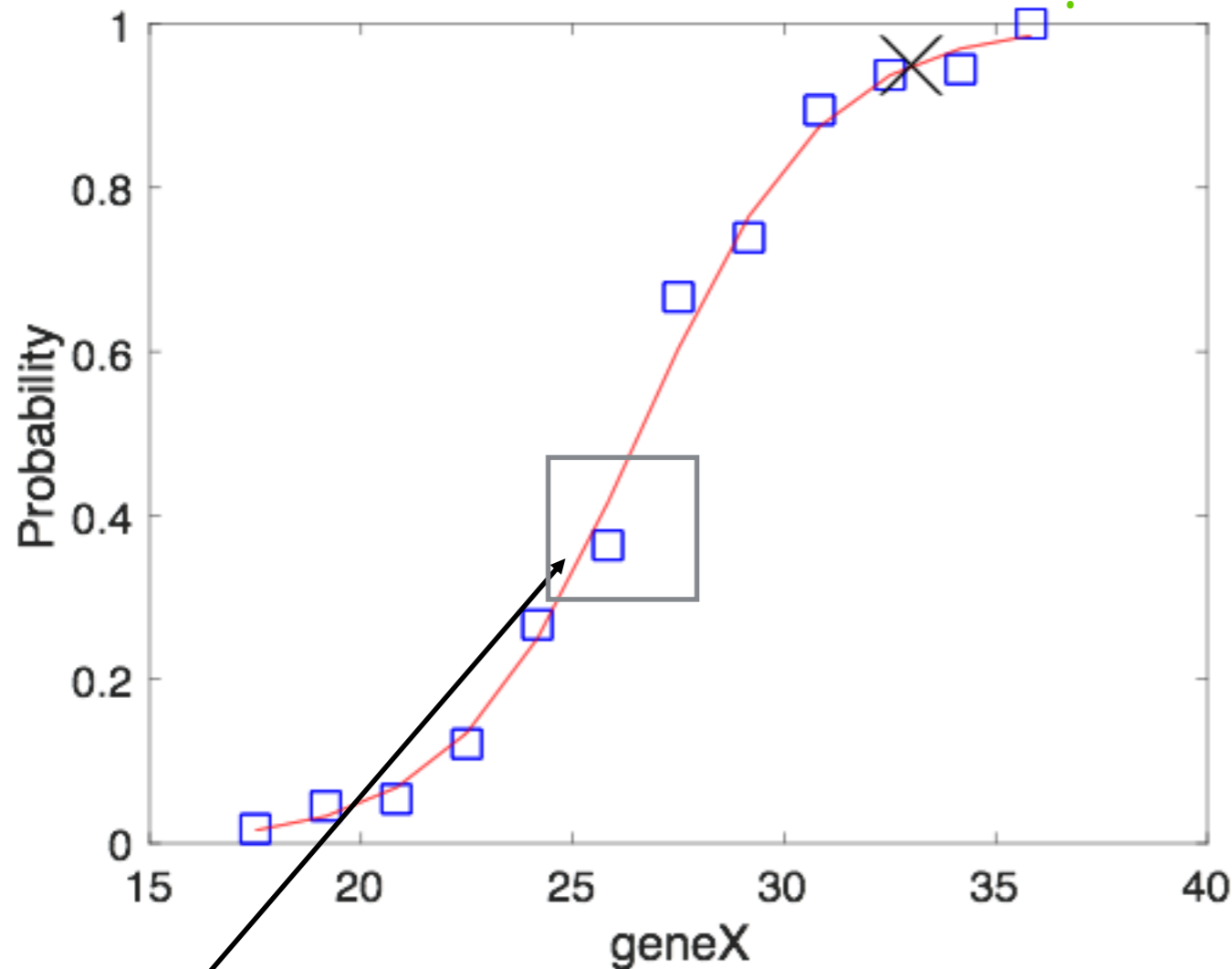
$$P(Y=1|\text{beta}) = \frac{\exp(b(1)+b(2)x)}{1+\exp(b(1)+b(2)x)}$$

```
% prediction by using logistic model
% given that patient has an average RNA level from isolated cells
genepredict=40

% what is the risk of having cancer?
% model equation
cancerriskpro=exp(logitCoef(1)+genepredict*logitCoef(2))/(1+exp(logitCoef(1)+genepredict
% probability
disp(cancerriskpro)
figure(3)
plot(geneX,proportion,'bs', geneX,logitFit,'r-', 'markersize',16);
hold on
plot(genepredict,cancerriskpro,'mo', 'markersize',34);
xlabel('geneX');
ylabel('Probability');
set(gca,'fontsize',18)
```



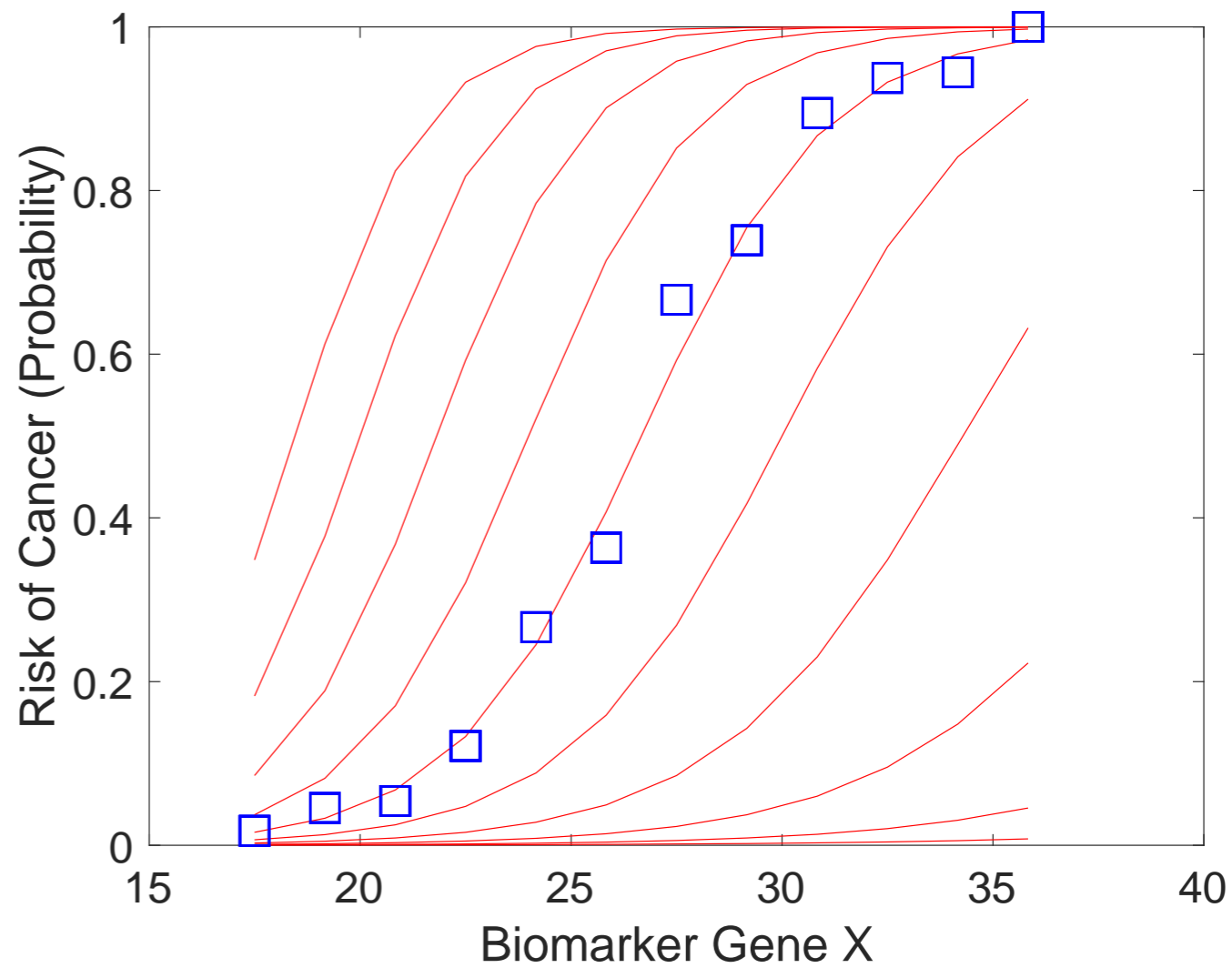
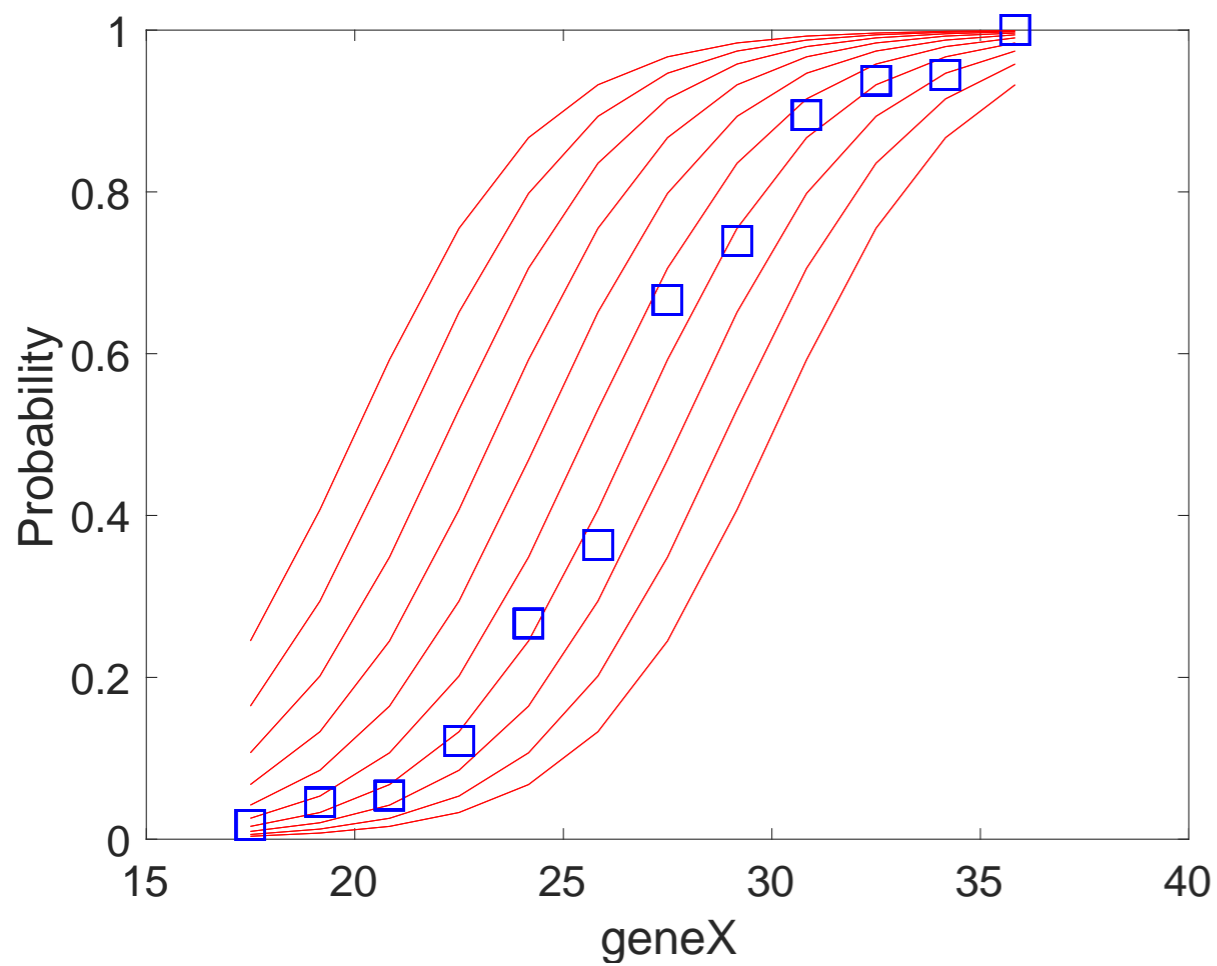
Coefficients are estimated by using a maximum likelihood estimation method where coefficients maximizes the prediction of observed values in the data



points on a line represents the highest points in the probability distribution

$$\log(\text{odds}) = b_0 + b_1x = -12.12 + 0.45x$$

The effect of coefficients on the shape of logistic model



Maximum likelihood estimation

Estimation parameters where the probability of observed data is maximized.

$$P(Y=1 | \text{coefficients}) = \frac{\exp(b(1) + b(2)x)}{1 + \exp(b(1) + b(2)x)}$$

Likelihood function

$$P(y | \beta(x)) = \frac{(\exp(b(1) + b(2)x))^y}{1 + \exp(b(1) + b(2)x)}$$

Computation of upper and lower limits of coefficients

$$b_1 \pm z \cdot SE_{b_1} = 0.38 \pm (1.96)(0.04) = X \pm Y$$

```
%Calculate the 95% confidence limits for the coefficients.
LL = stats.beta - 1.96.*stats.se;
UL = stats.beta + 1.96.*stats.se;

%
%Display the confidence intervals for the coefficients of the model for the relative risk of
% confidence interval for coefficient
%lower level
LL
%upperlevel
UL
```

```
LL =
    -15.1828
     0.3082

UL =
    -10.1668
     0.4651
```

Field	Value
beta	[-12.6748;0.3867]
dfc	10
sfit	0.5951
s	1
estdisp	0
covb	[1.6374,-0.0508;-0.0508,0....
se	[1.2796;0.0400]
coeffcorr	[1,-0.9907;-0.9907,1]
t	[-9.9053;9.6573]
p	[3.9472e-23;4.5767e-22]
resid	12x1 double
residp	12x1 double
residd	12x1 double
resida	12x1 double
wts	12x1 double

Now, odds of having cancer increase

$$(e^{b_1+z \cdot SE_{b_1}}, e^{b_1-z \cdot SE_{b_1}}) = (e^{0.30}, e^{.46}) = (1.36, 1.59)$$

```
%
exp(LL(2))
exp(UL(2))
```

```
ans =
    1.3610

ans =
    1.5922
```

$$\log(\text{odds}) = b_0 + b_1x = -12.12 + 0.38x$$

1. if $P = 1.2e-22$, we can reject the null hypothesis that $b_1 = 0$.

2. We use the estimate $b_1 = 0.45$ and its standard error $SE_{b_1} = 0.04$

to compute the 95% confidence interval for β_1 :

$$b_1 \pm z^*SE_{b_1} = \underline{0.38} \pm (1.96)(0.04) = X \pm Y$$

Our estimate of the slope is 0.38 and we are 95% confident that the true value is between 0.30 and 0.46.

For the odds ratio, the estimate is 1.56 and 95% confidence interval is

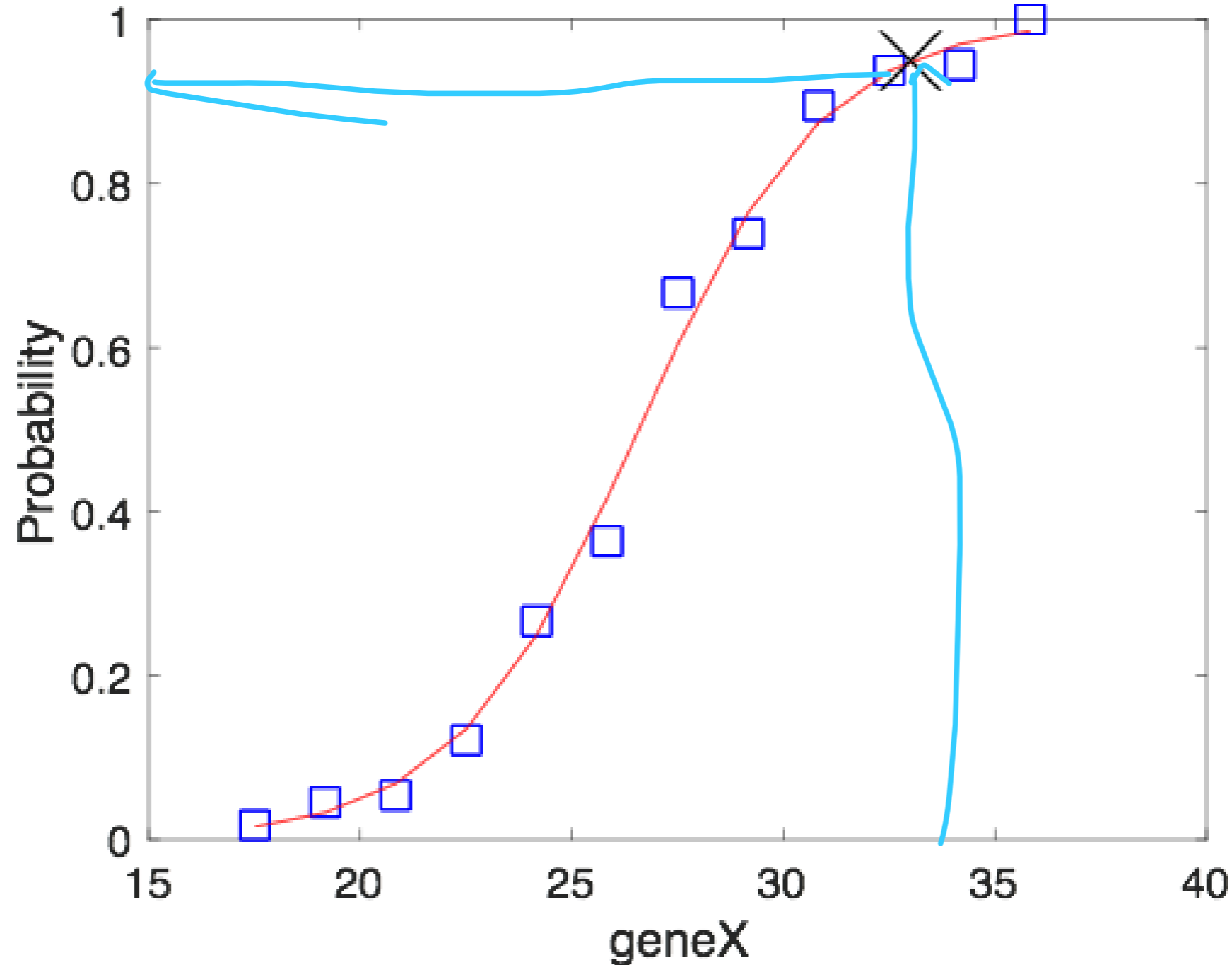
$$(e^{b_1+z^*SE_{b_1}}, e^{b_1-z^*SE_{b_1}}) = (e^{0.30}, e^{0.46}) = (1.36, 1.59)$$

```
%  
exp(LL(2))  
exp(UL(2))
```

The odds of having cancer increase by a factor of 1.56 for each unit increase in the log concentration of gene expression

Prediction of cancer risk with logistic model

Given that geneX results of patient is 33 ug/ml, what is the risk of having disease?

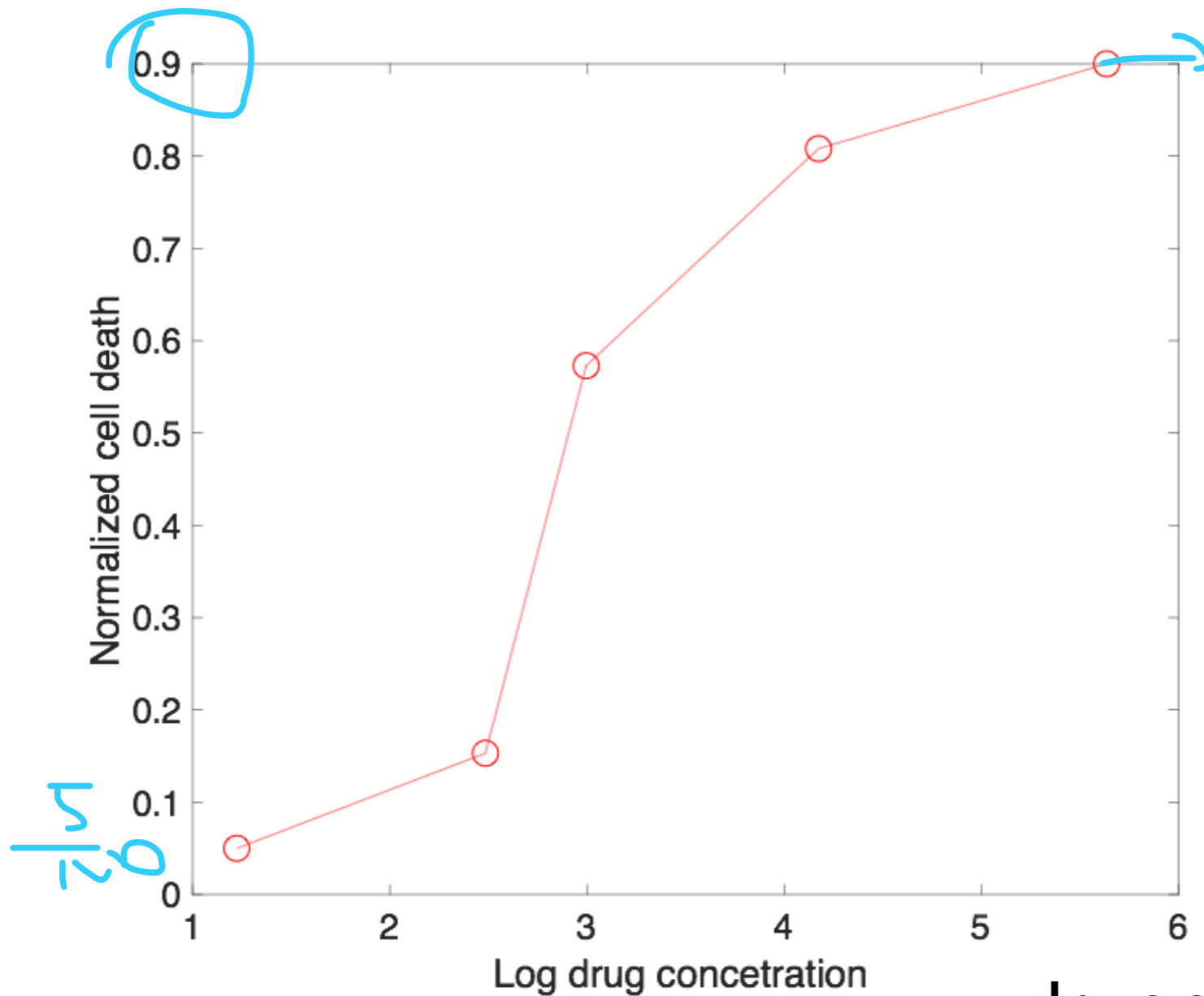


$\frac{P}{1+P} \rightarrow \text{odds}$

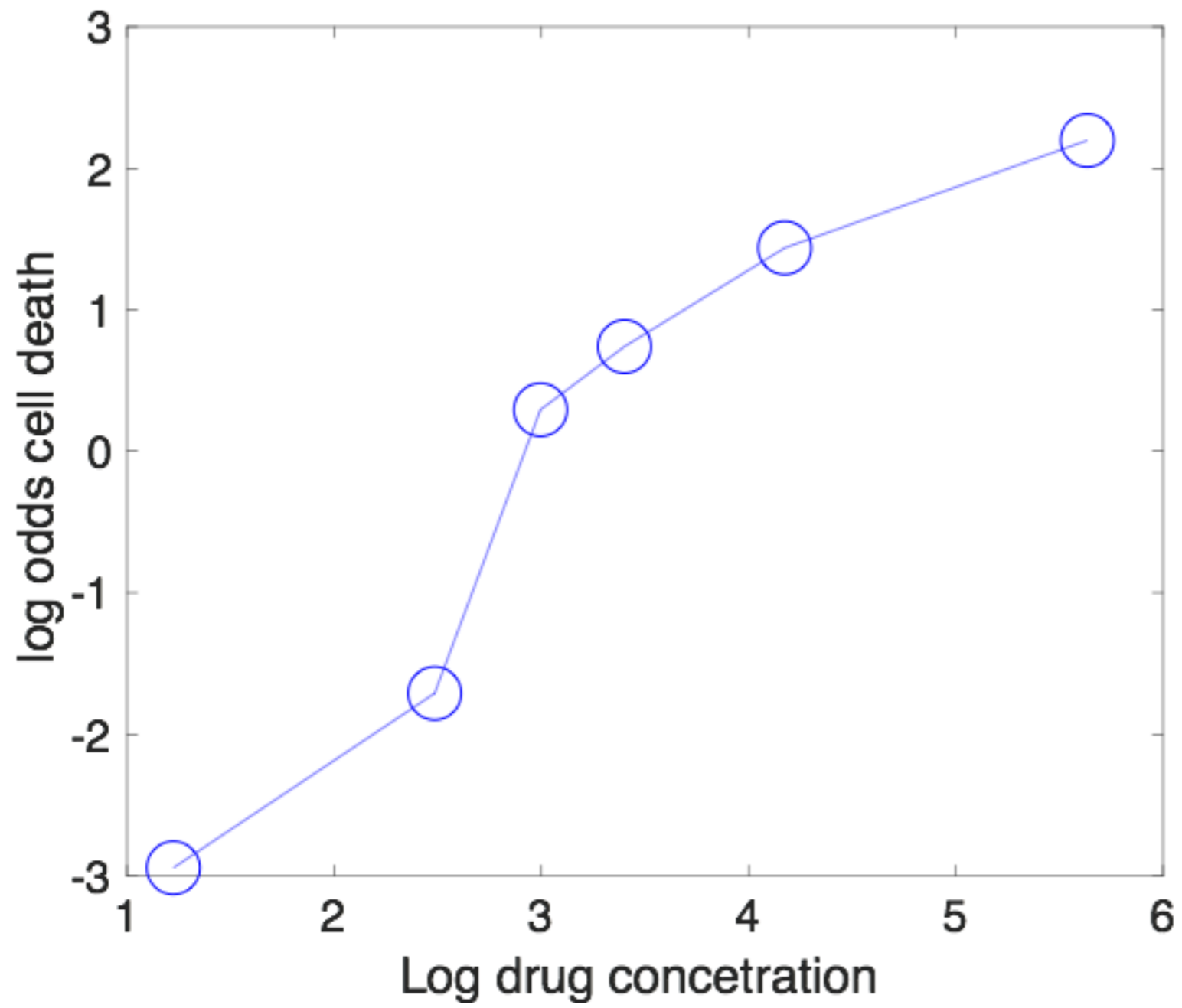
In this experiment we examine how well a drug kills cancer cells.

The explanatory variable is the log concentration of the drug.

We count each cells whether was either killed or alive.



drugconc=[3.4, 12, 20, 30, 65, 280]
cancercells=[100, 98, 96, 96, 99, 100]
numberkilled=[5, 15, 55, 65, 80, 90]



Interpretation of logistic regression results

$$\log(\text{odds}) = b_0 + b_1x = \underline{-18.64} + \underline{4.25}x$$

1. if $P = \underline{0.018}$, we can reject the null hypothesis that $b_1 = 0$.

2. We use the estimate $b_1 = 4.25$ and its standard error $\underline{SE_{b_1}} = 1.428$

to compute the 95% confidence interval for β_1 :

$$b_1 \pm z^*SE_{b_1} = 4.249 \pm (1.96)(1.428) = X \pm Y$$

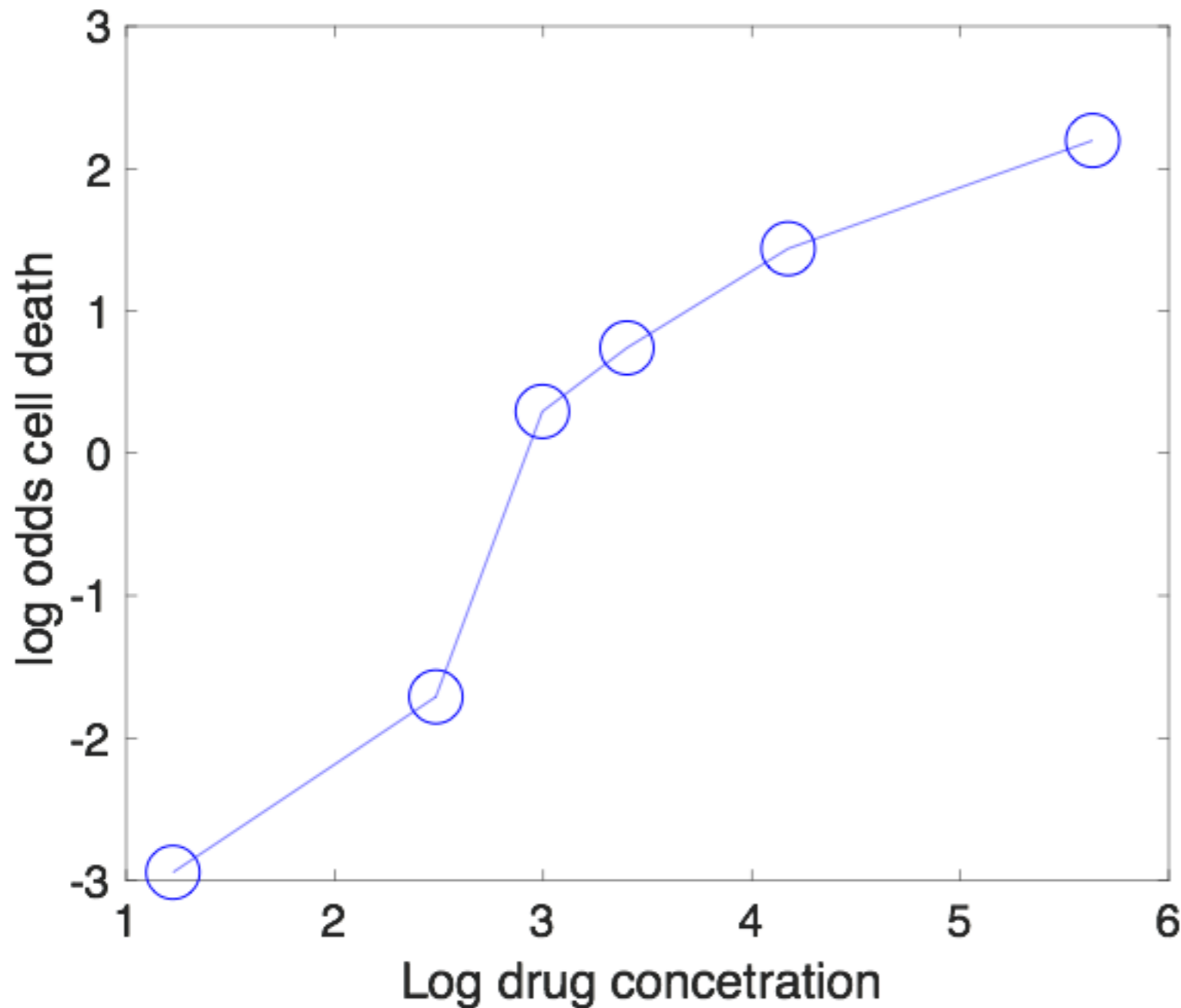
Our estimate of the slope is 4.25 and we are 95% confident that the true value is between 0.24 and 4.26.

For the odds ratio, the estimate is 9.48 and 95% confidence interval is

$$(e^{b_1+z^*SE_{b_1}}, e^{b_1-z^*SE_{b_1}}) = (e^{0.23588}, e^{4.26212}) = (1.27, 70.96)$$

Conclusion

The odds of killing cancer cells increase by a factor of 9.5 for each unit increase in the log concentration of drug




Multiple logistic regression

The data set includes three gene variables: geneA, geneB, and geneC.

We examined the model where geneA was used to predict the odds.

Do the other explanatory variables contain additional information that will give us a better prediction?

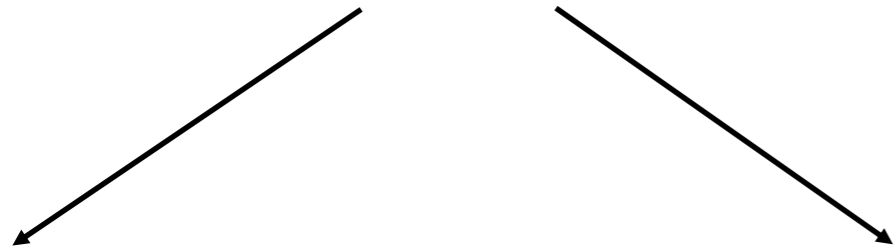
We use **multiple logistic regression**.

$$\begin{aligned}\log(\text{odds}) &= b_0 + b_1 \text{geneA} + b_2 \text{geneB} + b_3 \text{geneC} \\ &= -14.26 + 0.58 \text{geneA} + 0.68 \text{geneB} + 3.47 \text{geneC}\end{aligned}$$


Machine learning with Matlab

It teaches the computer to think like humans. The data is provided and interpret to build a model

Supervised learning



Classification

- Nearest Neighbor
- Naïve Bayes
- Support Vector machines
- Random Forest
- Neuronal Networks

Regression

- Linear Reg
- Logistic Reg
- Gaussian model

Unsupervised learning

- Kmeans
- Hidden markov model
- Hierarchical model

Machine learning for biology

AlphaFold

Article

Highly accurate protein structure prediction with AlphaFold

<https://doi.org/10.1038/s41586-021-03819-2>

Received: 11 May 2021

Accepted: 12 July 2021

Published online: 15 July 2021

Open access

Check for updates

John Jumper^{1,2,3}, Richard Evans^{1,4}, Alexander Pritzel^{1,4}, Tim Green^{1,4}, Michael Figurnov^{1,4}, Olaf Ronneberger^{1,4}, Kathryn Tunyasuvunakool^{1,4}, Russ Bates^{1,4}, Augustin Židek^{1,4}, Anna Potapenko^{1,4}, Alex Bridgland^{1,4}, Clemens Meyer^{1,4}, Simon A. A. Kohl^{1,4}, Andrew J. Ballard^{1,4}, Andrew Cowie^{1,4}, Bernardino Romera-Paredes^{1,4}, Stanislav Nikolov^{1,4}, Rishub Jain^{1,4}, Jonas Adler¹, Trevor Back¹, Stig Petersen¹, David Reiman¹, Ellen Clancy¹, Michal Zielinski¹, Martin Steinegger^{2,3}, Michalina Pacholska¹, Tamas Berghammer¹, Sebastian Bodenstein¹, David Silver¹, Oriol Vinyals¹, Andrew W. Senior¹, Koray Kavukcuoglu¹, Pushmeet Kohli¹ & Demis Hassabis^{1,4,5}

Proteins are essential to life, and understanding their structure can facilitate a mechanistic understanding of their function. Through an enormous experimental effort^{1–4}, the structures of around 100,000 unique proteins have been determined⁵, but this represents a small fraction of the billions of known protein sequences^{6,7}. Structural coverage is bottlenecked by the months to years of painstaking effort required to determine a single protein structure. Accurate computational approaches are needed to address this gap and to enable large-scale structural bioinformatics. Predicting the three-dimensional structure that a protein will adopt based solely on its amino acid sequence—the structure prediction component of the ‘protein folding problem’⁸—has been an important open research problem for more than 50 years⁹. Despite recent progress^{10–14}, existing methods fall far short of atomic accuracy, especially when no homologous structure is available. Here we provide the first computational method that can regularly predict protein structures with atomic accuracy even in cases in which no similar structure is known. We validated an entirely redesigned version of our neural network-based model, AlphaFold, in the challenging 14th Critical Assessment of protein Structure Prediction (CASP14)¹⁵, demonstrating accuracy competitive with experimental structures in a majority of cases and greatly outperforming other methods. Underpinning the latest version of AlphaFold is a novel machine learning approach that incorporates physical and biological knowledge about protein structure, leveraging multi-sequence alignments, into the design of the deep learning algorithm.

AlphaFold Protein Structure Database

Home About FAQs Downloads API

AlphaFold

Protein Structure Database

Developed by Google DeepMind and EMBL-EBI

Search for protein, gene, UniProt accession or organism or sequence search BETA Search

Examples:

[See search help](#) [Go to online course](#)

Machine learning for self driving cars

Neural Networks

Apply cutting-edge research to train deep neural networks on problems ranging from perception to control. Our per-camera networks analyze raw images to perform semantic segmentation, object detection and monocular depth estimation. Our birds-eye-view networks take video from all cameras to output the road layout, static infrastructure and 3D objects directly in the top-down view. Our networks learn from the most complicated and diverse scenarios in the world, iteratively sourced from our fleet of millions of vehicles in real time. A full build of Autopilot neural networks involves 48 networks that take 70,000 GPU hours to train. Together, they output 1,000 distinct tensors (predictions) at each timestep.

Autonomy Algorithms

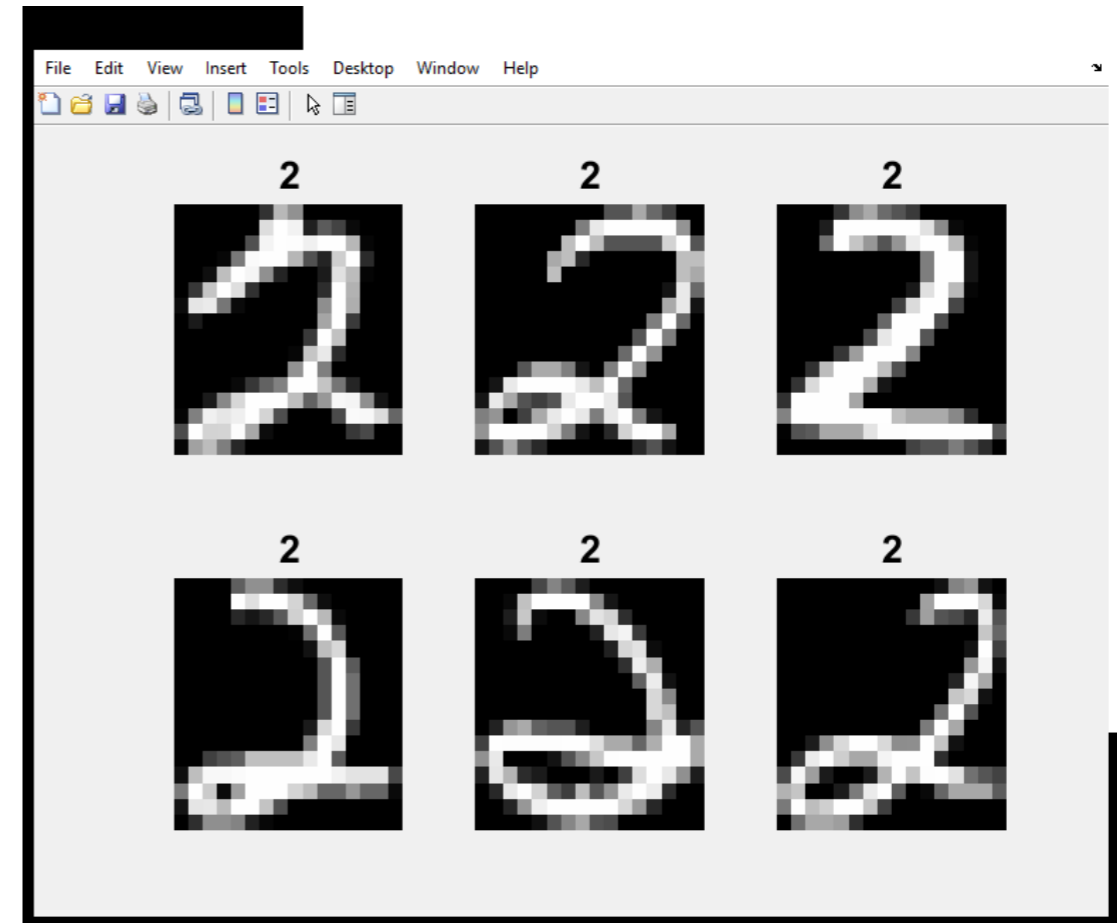
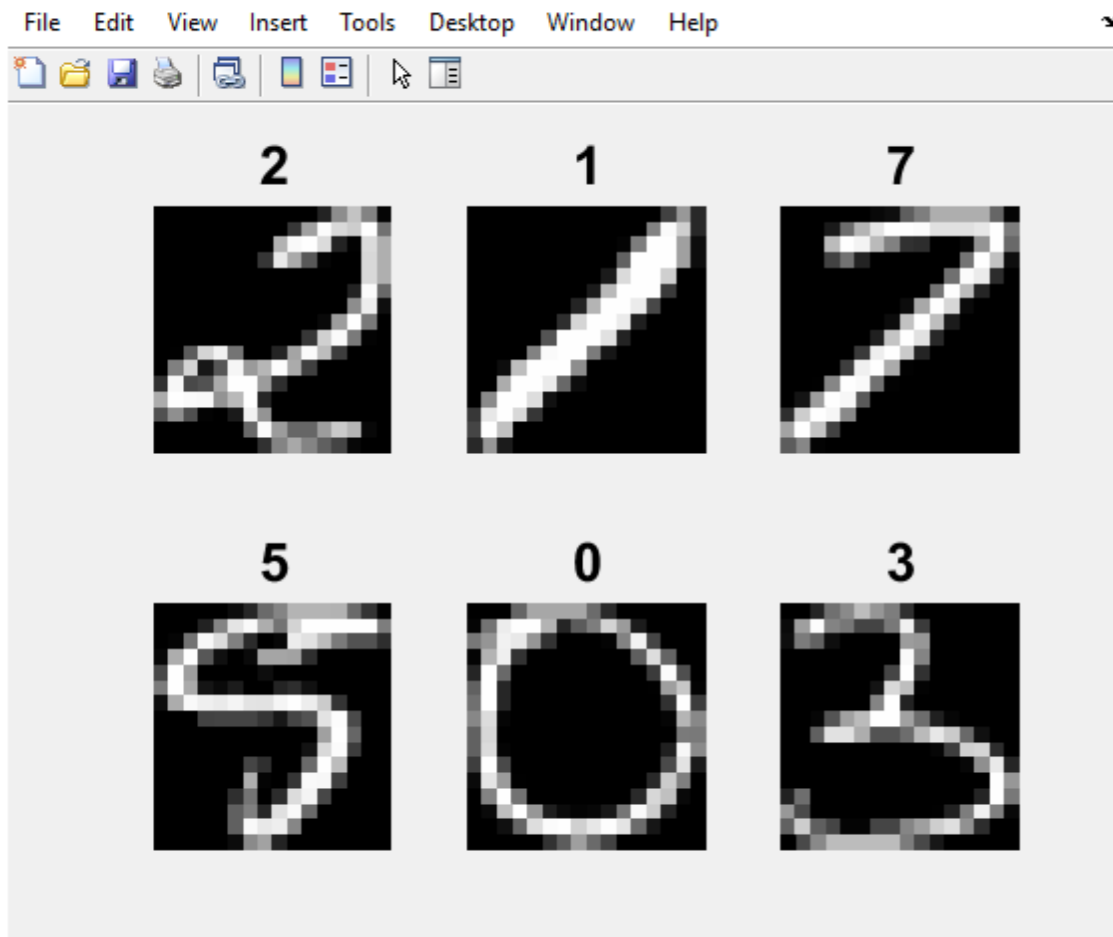
Develop the core algorithms that drive the car by creating a high-fidelity representation of the world and planning trajectories in that space. In order to train the neural networks to predict such representations, algorithmically create accurate and large-scale ground truth data by combining information from the car's sensors across space and time. Use state-of-the-art techniques to build a robust planning and decision-making system that operates in complicated real-world situations under uncertainty. Evaluate your algorithms at the scale of the entire Tesla fleet.



Example: Hand written recognition

Classic problem in machine learning

Problem: Can we teach the computer to read the hand written digits ?



Can you predict the following hand written digit? Is it 1 or 2?



Is it 1 or 2?

Labels

2

2

2

As we humans, computers also make mistakes!
How to reduce error rate?

1. Use many training samples
2. Use many features

Step 1: Convert the images into a linear form

11000x256 double

	70	71	72	73	74	75	76	77	78	79	80
1	0	0	0	39	216	255	245	98	3	0	
2	0	0	0	117	255	255	255	255	255	255	
3	0	0	0	0	0	27	231	255	255	114	
4	0	0	0	0	5	75	238	255	250	222	
5	0	0	0	0	11	215	224	40	0	0	
6	0	0	0	0	93	255	255	255	231	69	
7	0	0	64	103	255	255	255	255	255	255	
8	0	0	0	0	0	54	226	255	255	255	
9	0	0	0	0	0	99	255	255	194	9	
10	0	0	0	0	71	235	234	16	0	158	
11	0	0	0	0	19	163	252	255	229	70	
12	0	0	0	0	0	0	212	255	255	255	
13	0	0	0	0	0	48	230	255	254	112	
14	0	0	0	0	16	210	255	249	129	0	
15	0	0	0	16	154	255	255	156	13	0	
16	0	0	0	0	0	72	250	90	0	0	
17	0	0	0	0	17	218	255	255	91	0	
18	0	0	0	0	131	255	255	253	160	16	
19	255	255	255	255	255	255	255	255	255	249	
20	0	106	222	255	255	255	255	255	255	72	
21	0	0	0	0	67	214	229	91	0	0	
22	0	0	0	99	229	255	255	255	255	178	
23	0	68	189	255	255	255	255	255	255	255	
24	0	0	0	0	131	255	255	222	55	0	
25	255	255	255	221	162	162	83	0	0	0	

Command History

Step 2: Separate data into test and test set

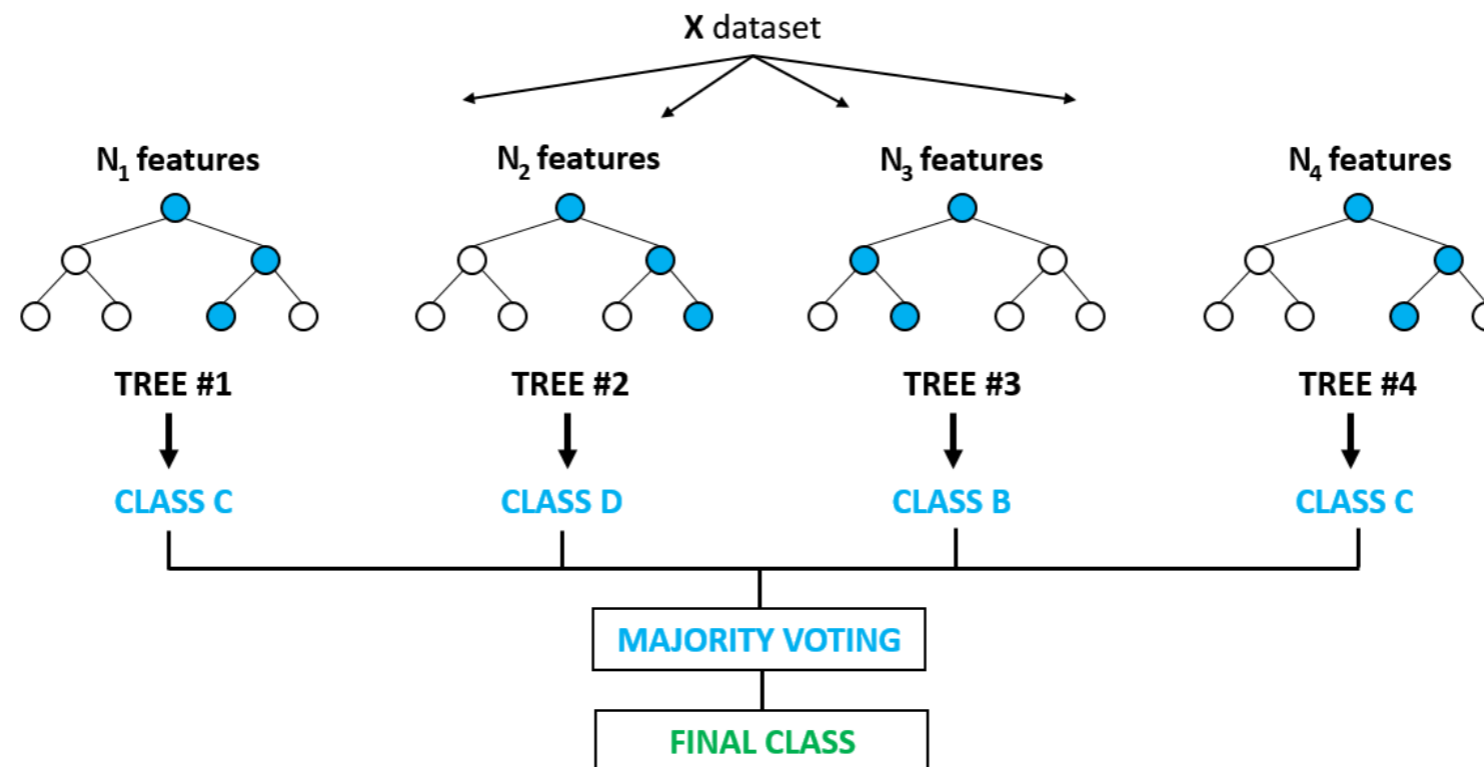
Step 2: Separate data into train and test set

```
366  
367  
368  
...  
369 %%  
370 - X=alldigilinear  
371 - cv = cvpartition(y, 'holdout', .5);  
372 - Xtrain = X(cv.training,:);  
373 - Ytrain = y(cv.training,1);  
374 - Xtest = X(cv.test,:);  
375 - Ytest = y(cv.test,1);  
376  
377  
378
```

Xtest	double	5500x256 double
Xtrain	double	5500x256 double
y	double	11000x1 double
ylabel	double	[1,2,3,4,5,6,7,8,9,0]
ypred	double	5500x1 double
Ytest	double	5500x1 double
Ytrain	double	5500x1 double

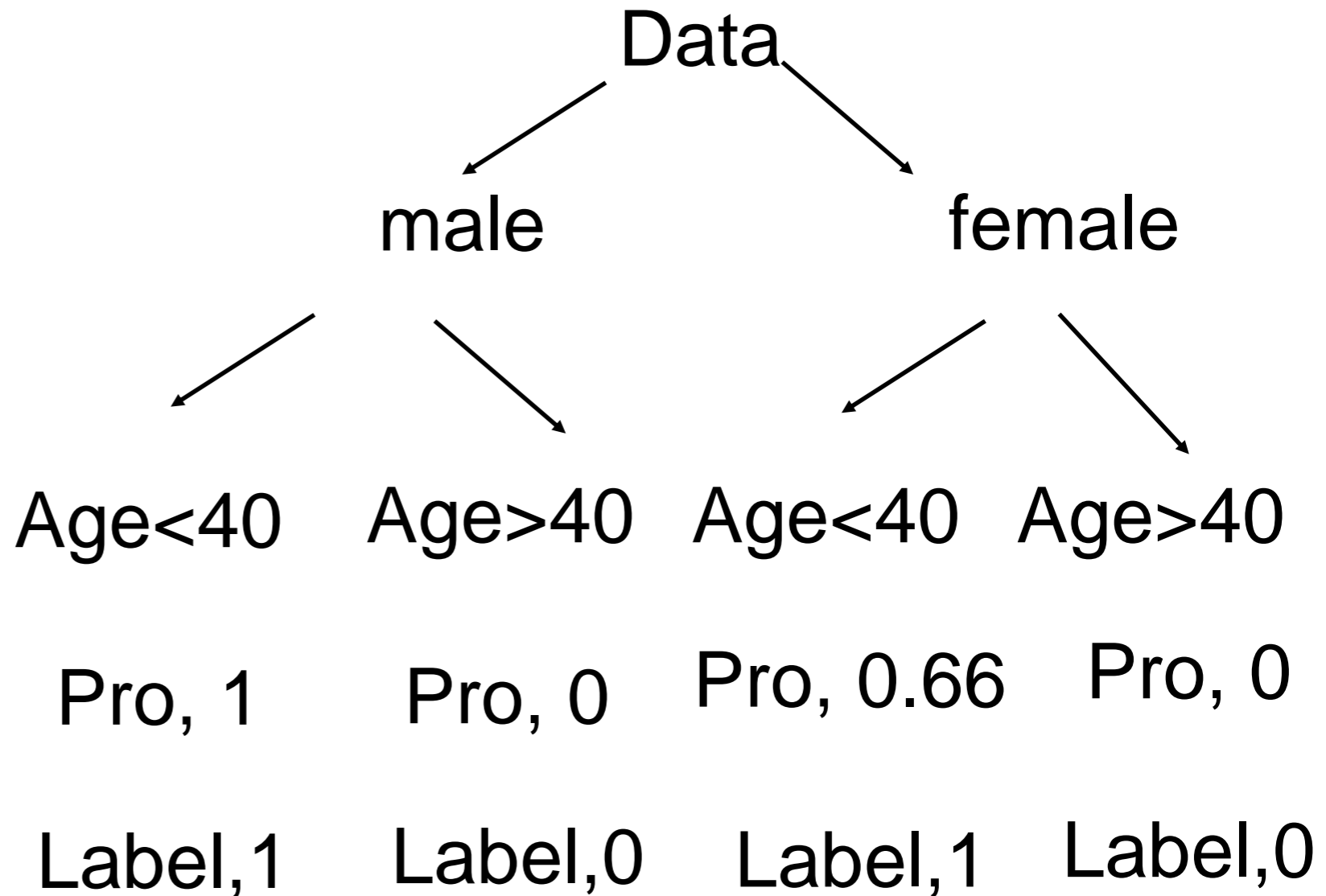
Classification Tree

- Used for multiclass classification.
- It is an iterative process for splitting data into partitions and split them further into branches
- The method based on finding features that splits data.
- We create a model that predicts the label of a target variable by learning decision rules extracted from the data features.



Build a simple Classification Tree for fail or pass the course

people	gender	Age <40		Pass or fail
1	1	1		1
2	1	1		1
3	1	0		0
4	1	1		0
5	0	1		1
6	0	0		0
7	0	1		1
8	0	0		0



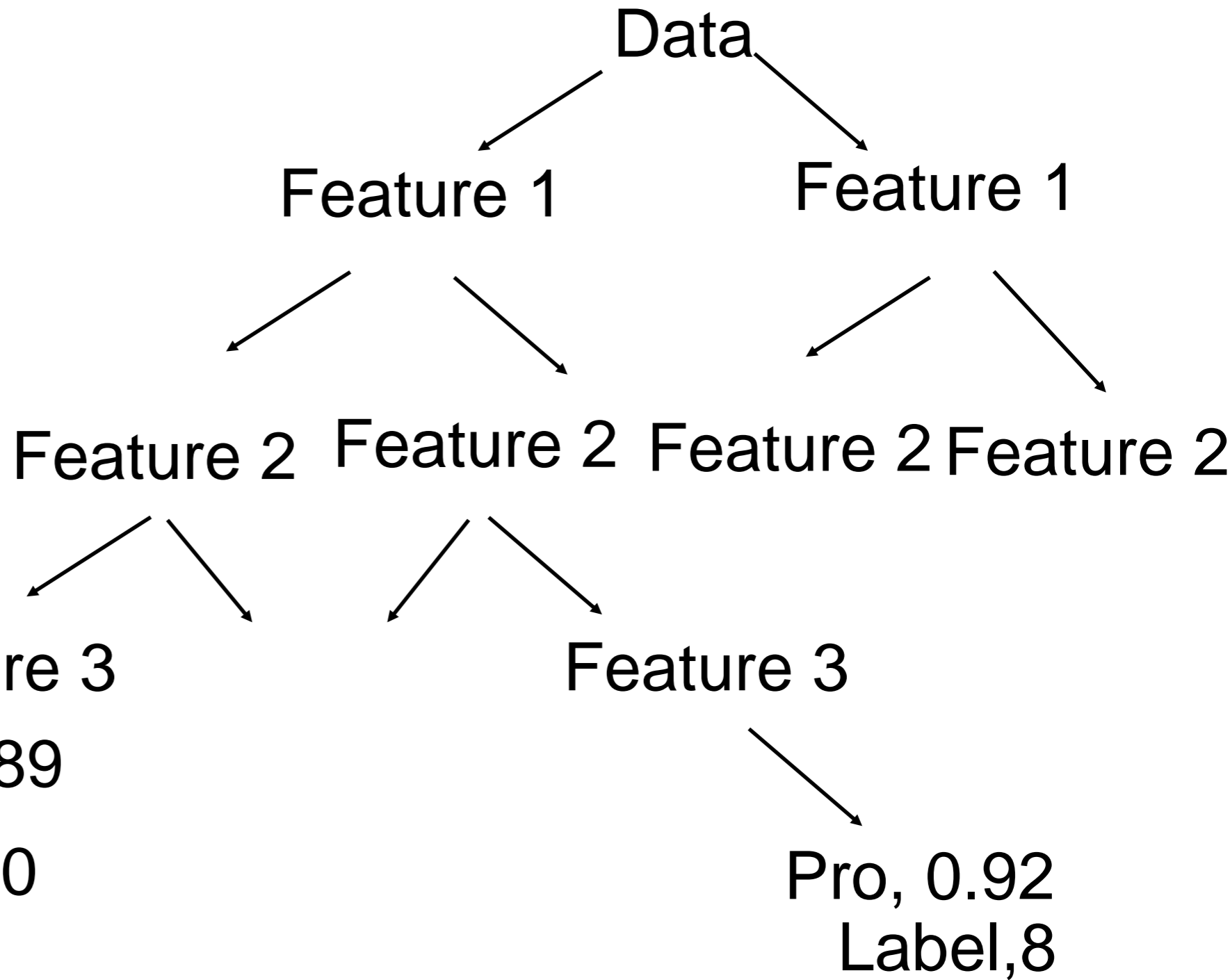
Test samples: a) male,
age>24
b) Female, age

Feature 1: Feature 2:
Female=1 Age<40=1
Male=0 Age>40=0

Features

11000x256 double

	70	71	72	73	74	75	76	77	78	79	80
1	0	0	0	39	216	255	245	98	3	0	
2	0	0	0	117	255	255	255	255	255	255	
3	0	0	0	0	0	27	231	255	255	114	
4	0	0	0	0	5	75	238	255	250	222	
5	0	0	0	0	11	215	224	40	0	0	
6	0	0	0	0	93	255	255	255	231	69	
7	0	0	64	103	255	255	255	255	255	255	
8	0	0	0	0	0	54	226	255	255	255	
9	0	0	0	0	0	99	255	255	194	9	
10	0	0	0	0	71	235	234	16	0	158	
11	0	0	0	0	19	163	252	255	229	70	
12	0	0	0	0	0	0	212	255	255	255	
13	0	0	0	0	0	48	230	255	254	112	
14	0	0	0	0	16	210	255	249	129	0	
15	0	0	0	16	154	255	255	156	13	0	
16	0	0	0	0	0	72	250	90	0	0	
17	0	0	0	0	17	218	255	255	91	0	
18	0	0	0	0	131	255	255	253	160	16	
19	255	255	255	255	255	255	255	255	255	249	
20	0	106	222	255	255	255	255	255	255	72	
21	0	0	0	0	67	214	229	91	0	0	
22	0	0	0	99	229	255	255	255	255	178	
23	0	68	189	255	255	255	255	255	255	255	
24	0	0	0	0	131	255	255	222	55	0	
25	255	255	255	221	162	162	83	0	0	0	



Compare predicted and true labels

```
%%
% Train and Predict Using a Single Classification Tree
mdl_ctree = ClassificationTree.fit(Xtrain,Ytrain);
ypred = predict(mdl_ctree,Xtest);
Confmat_ctree = confusionmat(Ytest,ypred);

%
% Train and Predict Using Bagged Decision Trees
mdl = fitensemble(Xtrain,Ytrain,'bag',200,'tree','type','Classification');
ypred = predict(mdl,Xtest);
Confmat_bag = confusionmat(Ytest,ypred);
```

File Edit View Insert Tools Desktop Window Help

Confusion Matrix: Single Classification Tree

True Class \ Predicted Class	1	2	3	4	5	6	7	8	9	10
1	494	4	7	8	9		17	2	8	1
2	3	484	15	2	10	14	9		9	4
3	14	11	416	20	12	16	13	14	29	5
4	12	5	17	438	4	26	5	11	22	10
5	7	12	14	6	460	19	5	10	4	13
6	5	5	14	56	17	420	7	3	15	8
7	8	10	22	2	21	9	467		11	
8		6	20	11	13	2		473	10	15
9	13	11	47	42	14	19	6	7	365	26
10	2	4	4	9	30	6		14	17	464

File Edit View Insert Tools Desktop Window Help

Confusion Matrix: Ensemble of Classification Trees

True Class \ Predicted Class	1	2	3	4	5	6	7	8	9	10
1	541	2					5		2	
2		544			3		3			
3	2		524	1	3	1	7	4	5	3
4	1		8	525		6	2	2	3	3
5			3		538		2		1	6
6	1	1		12	2	530	2			2
7	1	5	3		3		538			
8		1			4			538	2	5
9	1	4	8	6	2	6	2		506	15
10		1	1	1	6			6	3	532

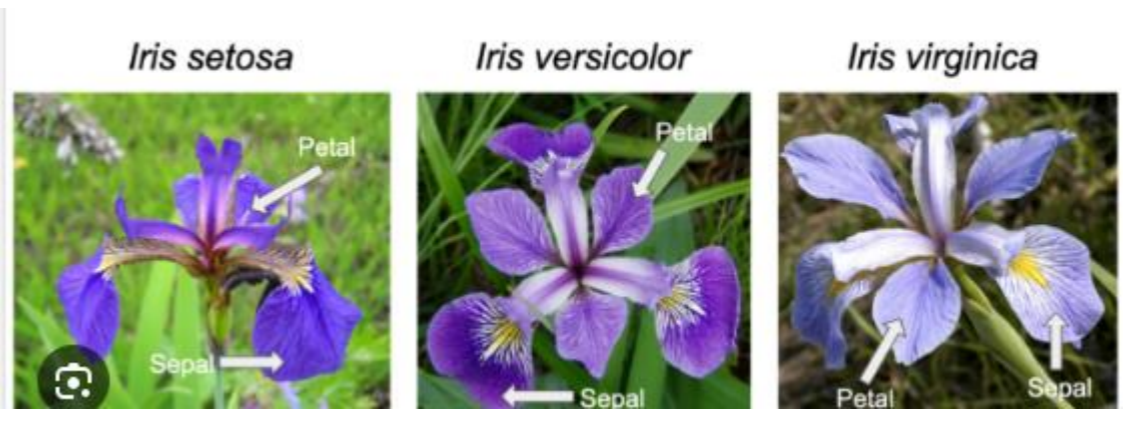
Other examples for decision tree

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)



Iris setosa



Iris versicolor



Iris virginica



File Tools Desktop Tree Window Help



Click to display:

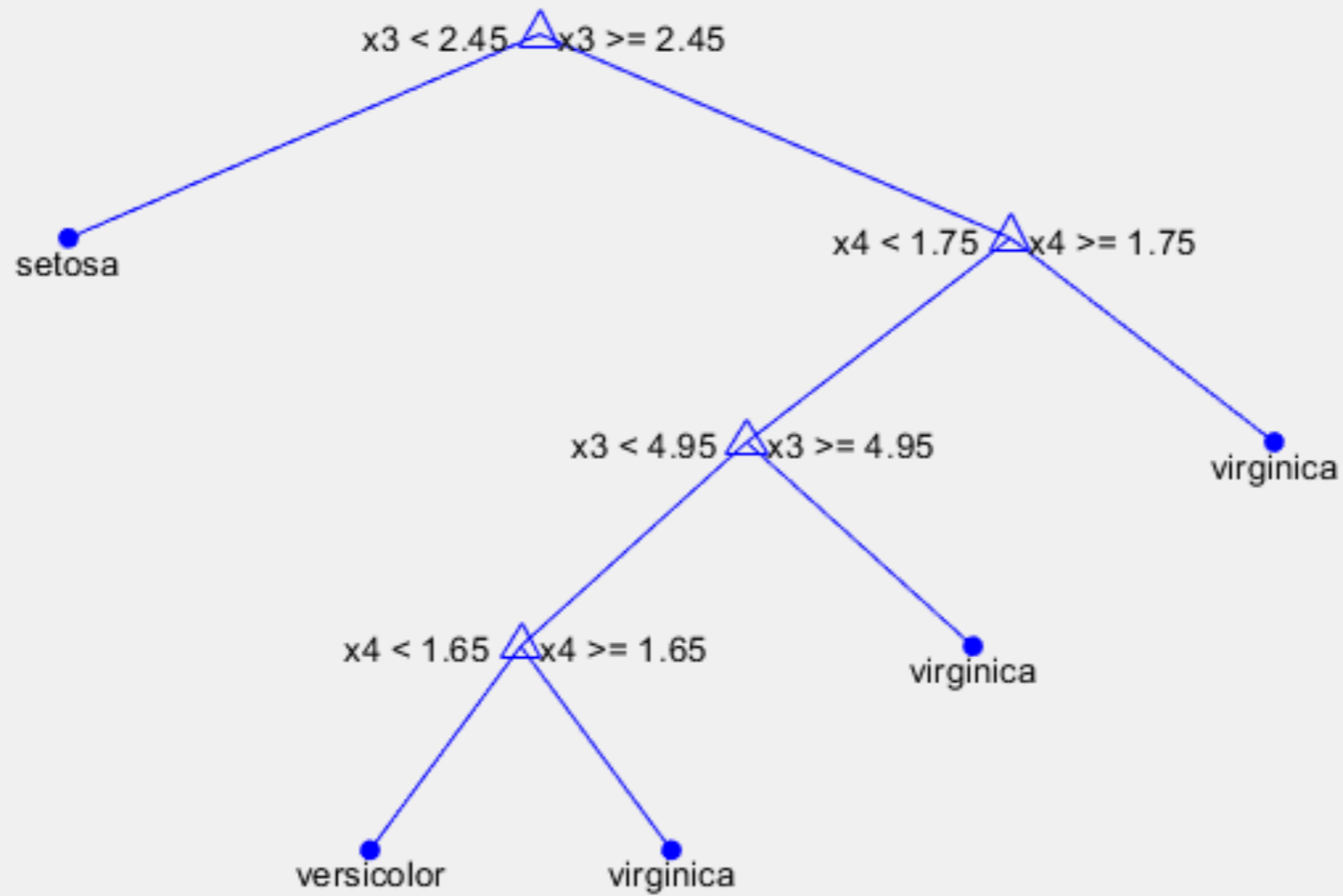
Identity

Magnification:



100%


Pruning level:

0 of 4



Adaptive tracking algorithm for trajectory analysis of cells and layer-by-layer assessment of motility dynamics

Mohammad Haroon Qureshi^{a,b}, Nurhan Ozlu^a, Halil Bayraktar^c  

Show more 

 Add to Mendeley  Share  Cite

<https://doi.org/10.1016/j.combiomed.2022.106193>

[Get rights and content](#) 

Abstract

Tracking biological objects such as cells or subcellular components with time-lapse microscopy enables us to understand the nature about the dynamics of cell behaviors. However, automatic object segmentation and extracting trajectories remain as a rate-limiting intrinsic challenges of video processing. This paper presents an adaptive tracking algorithm (Adtari) that automatically finds the optimal

