

Python and Artificial intelligence in Engineering

Ali Pilehvar Meibody
alipilehvar1999@gmail.com
ai.course22.alipilehvar@gmail.com

برنامه نویسی چیست

زبان‌های برنامه‌نویسی ساختارهای زبانی دستور مدار الگوریتم را به وسیلهٔ ساختارهای دستوری متفاوت برای اجرای رایانه توصیف کرد و با این روش امکان نوشتن برنامه جهت تولید نرم‌افزارهای جدید برای وارد کردن متن برنامه، اجرا، همگردادن و رفع اشکال آن هستند.

یک زبان برنامه‌نویسی یک زبان مصنوعی انگلیسی است که برای بیان محاسباتی که توسط یک ماشین (مخصوصاً رایانه) قابل انجام است، طراحی شده است. زبان‌های برنامه‌نویسی برای ایجاد برنامه‌هایی به کار می‌روند که رفتار یک ماشین را مشخص می‌کنند، الگوریتم دقیق را بیان می‌کنند، یا روشی برای تعامل انسان و رایانه هستند.



زبان برنامه نویسی پایتون Python

پایتون یک زبان برنامه نویسی شیء‌گرا، تفسیری، سطح بالا، و همه منظوره است، که خیدو فان روسوم [a] آن را طراحی کرده است، و اولین بار در سال ۱۹۹۱ منتشر شده است. فلسفه اصلی طراحی پایتون «خوانایی بالای کد» است و نویسه‌های فاصله خالی در آن معنادار هستند و مکرر استفاده می‌شوند. ساختار زبانی و دیدگاه شیء‌گرا در پایتون به گونه‌ای طراحی شده است که به برنامه نویس امکان نوشتن کد منطقی و واضح (بدون ابهام) را برای پروژه‌های کوچک و بزرگ می‌دهد

کاربردهای پایتون

هوش مصنوعی و یادگیری ماشین

حوزه تحلیل داده و علم داده

توسعه وب

توسعه بازی و بازی سازی

اتوماسیون

نرم افزار

شبکه

امنیت و هک

اینترنت اشیا



کتابخانه نامپای

NumPy یک کتابخانه برای محاسبات علمی در پایتون است. این کتابخانه نوعی محیط محاسباتی آرایه گرا را فراهم می‌کند که امکان دستکاری سریعتر و کارآمدتر مجموعه داده‌های بزرگ را فراهم خواهد کرد. کتابخانه نامپای همچنین طیف گسترده‌ای از توابع و عملیات ریاضی را ارائه می‌دهد که آن را به یک انتخاب عالی برای دانشمندان داده تبدیل می‌کند.

از مزایای کتابخانه NumPy موارد زیر را می‌توان نام برد:

NumPy یک کتابخانه قدرتمند برای محاسبات علمی در پایتون است. این عملیات سریع و کارآمد را روی آرایه‌ها و ماتریس‌های چندبعدی ارائه می‌دهد. همچنین بسیار بهینه شده است و بسیاری از توابع مفید برای جبر خطی، تبدیل فوریه و تولید اعداد تصادفی دارد.

از معایب کتابخانه NumPy موارد زیر را می‌توان نام برد:

استفاده از کتابخانه NumPy برای مبتدیان ممکن است دشوار باشد، زیرا به درک بالایی از جبر خطی و عملیات ماتریس نیاز دارد. علاوه بر این، عملکرد آن می‌تواند در هنگام برخورد با مجموعه داده‌های بزرگ کند شود.



کتابخانه پandas

Pandas یکی دیگر از قدرتمندترین کتابخانه های پایتون برای دستکاری و تجزیه و تحلیل دادهها است. پandas طیف وسیعی از ساختارها و ابزارهای داده را برای کار با دادهها فراهم میکند. همچنین مجموعهای قدرتمند از توابع برای مصورسازی و تجزیه و تحلیل دادهها را فراهم کرده که آن را به گزینهای عالی برای کارهای پروژه محور علم داده تبدیل خواهد کرد

از مزایای کتابخانه پandas موارد زیر را میتوان نام برد:

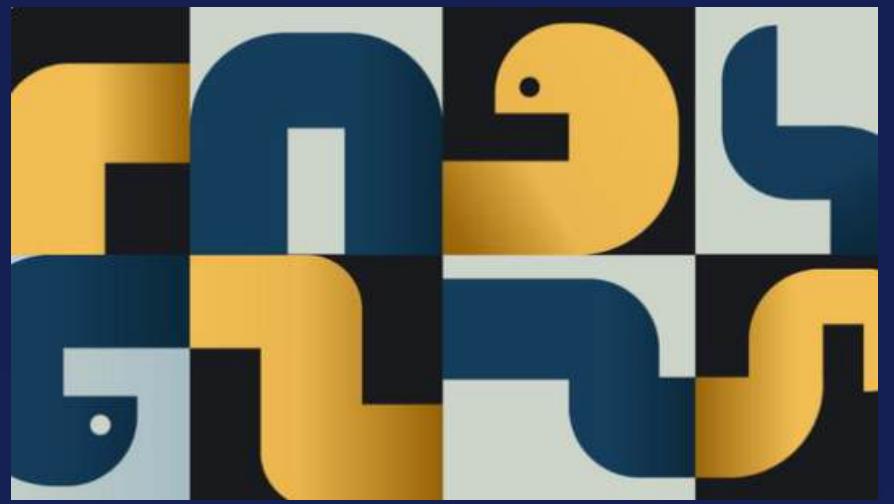
- یک کتابخانه قدرتمند تجزیه و تحلیل داده برای پایتون است.
- طیف وسیعی از ساختارها و ابزارهای داده را برای کار با دادهها فراهم میکند.
- این کتابخانه بسیار سریع، انعطاف‌پذیر و استفاده از آن بسیار آسان است.

از معایب کتابخانه پandas موارد زیر را میتوان نام برد:

- پandas هنگام برخورد با مجموعه دادههای بزرگ میتوانند کند باشد.
- درک نحو آن برای مبتدیان دشوار است.

کتابخانه مت پلات لیب matplotlib

Matplotlib نیز کتابخانه‌ای محبوب برای امور طراحی و گرافیکی در پایتون است. Matplotlib طیف گسترده‌ای از توابع و ویژگی‌های رسم را فراهم می‌کند و به دانشمندان داده اجازه می‌دهد تا تصاویر زیبایی از داده‌های خود ایجاد کنند. Matplotlib همچنین بسیار قابل تنظیم است و به دانشمندان داده اجازه می‌دهد تا نمودارهای خود را مطابق با نیازهای دقیق خود سفارشی کنند و می‌توان نام آن را بدون شک در بین فهرست بهترین کتابخانه‌های پایتون آورد.



محیط توسعه یکپارچه

Integrated Development Environment (IDE)

ابزارهای محیط توسعه یکپارچه یا همان IDE‌ها نرم افزارهایی هستند که امکاناتی را برای برنامه نویسان جهت کدنویسی، ساخت و توسعه برنامه‌ها و اپلیکیشن‌های دیگر فراهم می‌کنند. IDE برای دربرگرفتن تمام امکانات لازم برای پیاده‌سازی همه وظایف برنامه نویسی در قالب یک نرم افزار کاربردی طراحی شده است



SPYDER

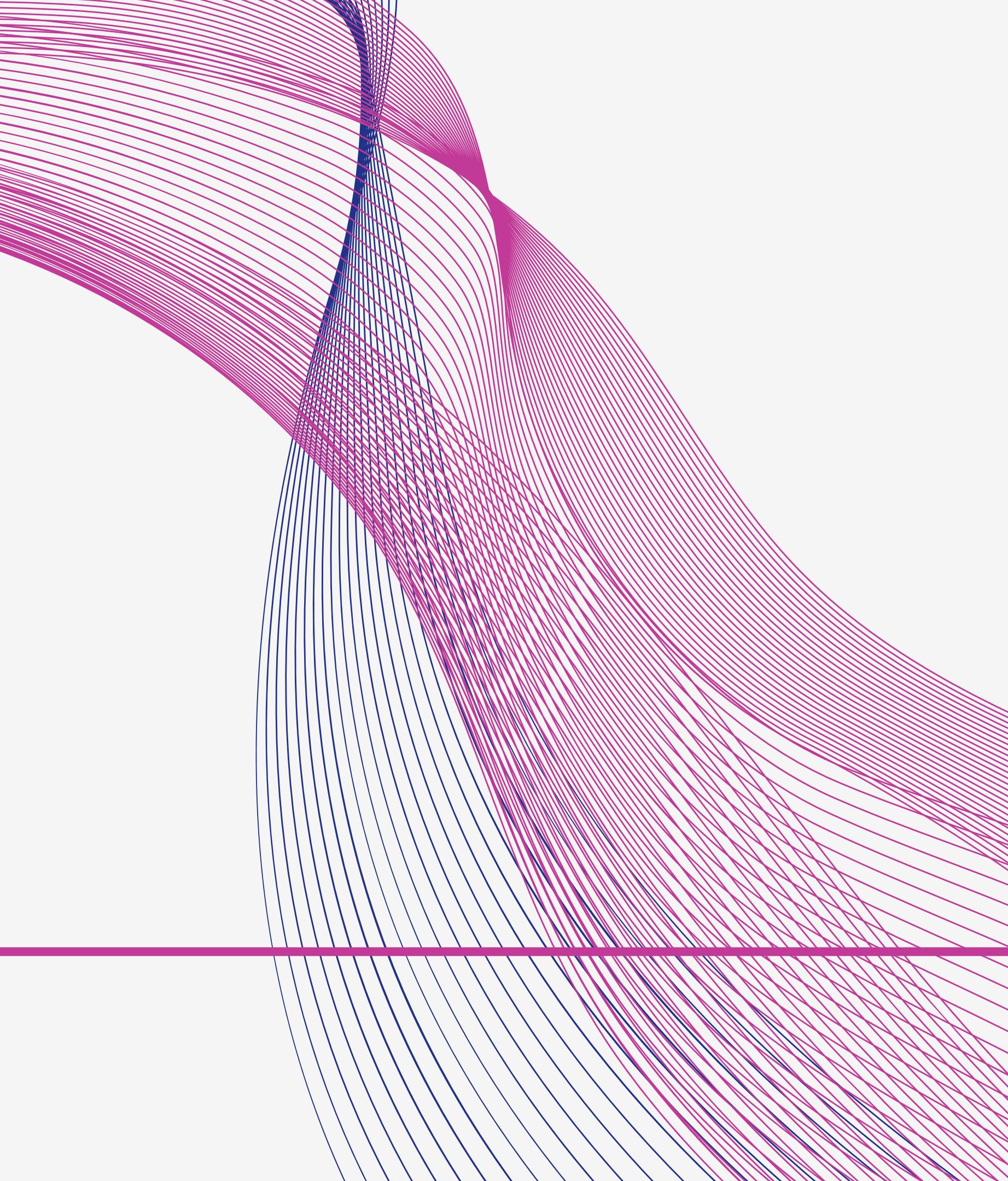


2: Introduction to python

Ali Pilehvar Meibody
alipilehvar1999@gmail.com
ai.course22.alipilehvar@gmail.com

خلاصه جلسه اول

- 1- تصورمان از یک کامپیوتر ، یک جسم جامد باشد (نهایت یک ماشین حساب)
- 2- منطق صفر و یک (زبان ماشینی)
- 3- زبان برنامه نویسی زبانی بین زبان ماشین و زبان انسان است
- 4- این زبان مانند هر زبان یکسری کلمات (وکب) و یکسری دستورات (گرامر) دارد
- 5- معرفی آنالوگوندا (مدیریت فایل های کامپیوترا)
- 6- معرفی محیط توسعه یکپارچه و اسپایدر
- 7- آشنایی با محیط اسپایدر





زبان برنامه نویسی پایتون Python

زبان پایتون زبان ماشین (صفر و یک) و زبان انسان (زبان انگلیسی) هست و این زبان مانند هر زبانی (انگلیسی , ایتالیایی , اسپانیایی ,...) دارای دو جز هست

جز اول : کلمات

جز دوم: قواعد و گرامر

محیط برنامه نویسی (اسپایدر..) مانند یک صفحه چت میتوانید که شما میتوانید کد خود را که به زبانی غیر از فارسی (زبان پایتون) هست بنویسید و تا زمانی که اجرایش نکنید ثابت هست.

زمانی که دکمه ران را زدید کرنل اسپایدر مانند یک مترجم از شروع (بالا) تا پایان کد شمارا ترجمه میکند (تبدیل به زبان ماشینی میکند) و با این تبدیل به ماشین دستورات خود را میدهید و ماشین آن را انجام میدهد.

این مترجم (کرنل اسپایدر) فقط یک زبان را بلد هست و آن پایتون هست ، شما میتوانید در صفحه اسپایدر (صفحه چت) هر حروفی را بنویسید اما فقط حروف و کلمات و قواعدی را میتواند ترجمه کند که به زبان پایتون نوشته شده است

زبان پایتون از دو جر تشکیل شده است
1- کلمات (keyword) و توابع داخلی (built-in function)
2- قواعد و دستور ها

ادامه درس :
متغیر ها (ظرفی برای ذخیره داده ها)
انواع متغیر ها (اعداد , رشته ها و ...)
محاسبه گر ها
تابع داخلی (open, print, input)
رشته ها
ست ها
لیست ها
تایل ها
دیکشنری ها

دستورات (IF, FOR, WHILE)

تابع (FUNCTION)

انواع متغیر ها

نوع : Text
عددی دنباله دار Numeric Types:
نقشه دار Sequence
ست Set Types: set,
بولیان Boolean Type:
بایتی Binary Types:
هیچ None Type:

Type: str
int, float, complex
Types: list, tuple, range
dict
frozenset
bool
bytes, bytearray, memoryview
NoneType

عملگر ها

** % / * - +

حساب: جمع | تفريق | ضرب | تقسيم | باقى مانده | به توان

مقاييسه: برابر == | غير برابر != | بزرگتر > | كوچکتر < | بزرگتر و مساوى = | كوچکتر و مساوى =<

منطقى: | اگر هر دو باهم صحيح باشد and | اگر يكى از آن ها صحيح باشد or

تشخيص: is , is not

عضویت: in , in not

تابع های داخلی

bool()	تبدیل به بولیان
int()	تبدیل به اینت (عدد)
float()	تبدیل به عدد اعشاری
complex()	تبدیل به عدد مختلط
str()	تبدیل به رشته
list()	ساخت لیست
input()	گرفتن ورودی از کاربر
print()	چاپ و نمایش
open()	باز کردن یک فایل
len()	اندازه و طول یک متغیر
range()	ساخت اعداد در یک محدوده
type()	نمایش تایپ و نوع متغیر
abs()	نمایش مطلق آن عدد
max()	پیدا کردن بزرگترین
min()	پیدا کردن کوچک ترین
sum()	جمع کل
pow()	توان

تابع های داخلی رشته ها

capitalize() Converts the first character to upper case

casefold() Converts string into lower case

center() Returns a centered string

count() Returns the number of times a specified value occurs in a string

encode() Returns an encoded version of the string

endswith() Returns true if the string ends with the specified value

expandtabs() Sets the tab size of the string

find() Searches the string for a specified value and returns the position of where it was found

format() Formats specified values in a string

format_map() Formats specified values in a string

index() Searches the string for a specified value and returns the position of where it was found

isalnum() Returns True if all characters in the string are alphanumeric

isalpha() Returns True if all characters in the string are in the alphabet

isdecimal() Returns True if all characters in the string are decimals

isdigit() Returns True if all characters in the string are digits

isidentifier() Returns True if the string is an identifier

islower() Returns True if all characters in the string are lower case

isnumeric() Returns True if all characters in the string are numeric

isprintable() Returns True if all characters in the string are printable

isspace() Returns True if all characters in the string are whitespaces

istitle() Returns True if the string follows the rules of a title

isupper() Returns True if all characters in the string are upper case

تابع های داخلی رشته ها

join() Joins the elements of an iterable to the end of the string

ljust() Returns a left justified version of the string

lower() Converts a string into lower case

lstrip() Returns a left trim version of the string

maketrans() Returns a translation table to be used in translations

partition() Returns a tuple where the string is parted into three parts

replace() Returns a string where a specified value is replaced with a specified value

rfind() Searches the string for a specified value and returns the last position of where it was found

rindex() Searches the string for a specified value and returns the last position of where it was found

rjust() Returns a right justified version of the string

rpartition() Returns a tuple where the string is parted into three parts

rsplit() Splits the string at the specified separator, and returns a list

rstrip() Returns a right trim version of the string

split() Splits the string at the specified separator, and returns a list

splitlines() Splits the string at line breaks and returns a list

startswith() Returns true if the string starts with the specified value

strip() Returns a trimmed version of the string

swapcase() Swaps cases, lower case becomes upper case and vice versa

title() Converts the first character of each word to upper case

translate() Returns a translated string

upper() Converts a string into upper case

zfill() Fills the string with a specified number of 0 values at the beginnin

تابع های داخلی لیست ها

append() Adds an element at the end of the list

clear() Removes all the elements from the list

copy() Returns a copy of the list

count() Returns the number of elements with the specified value

extend() Add the elements of a list (or any iterable), to the end of the current list

index() Returns the index of the first element with the specified value

insert() Adds an element at the specified position

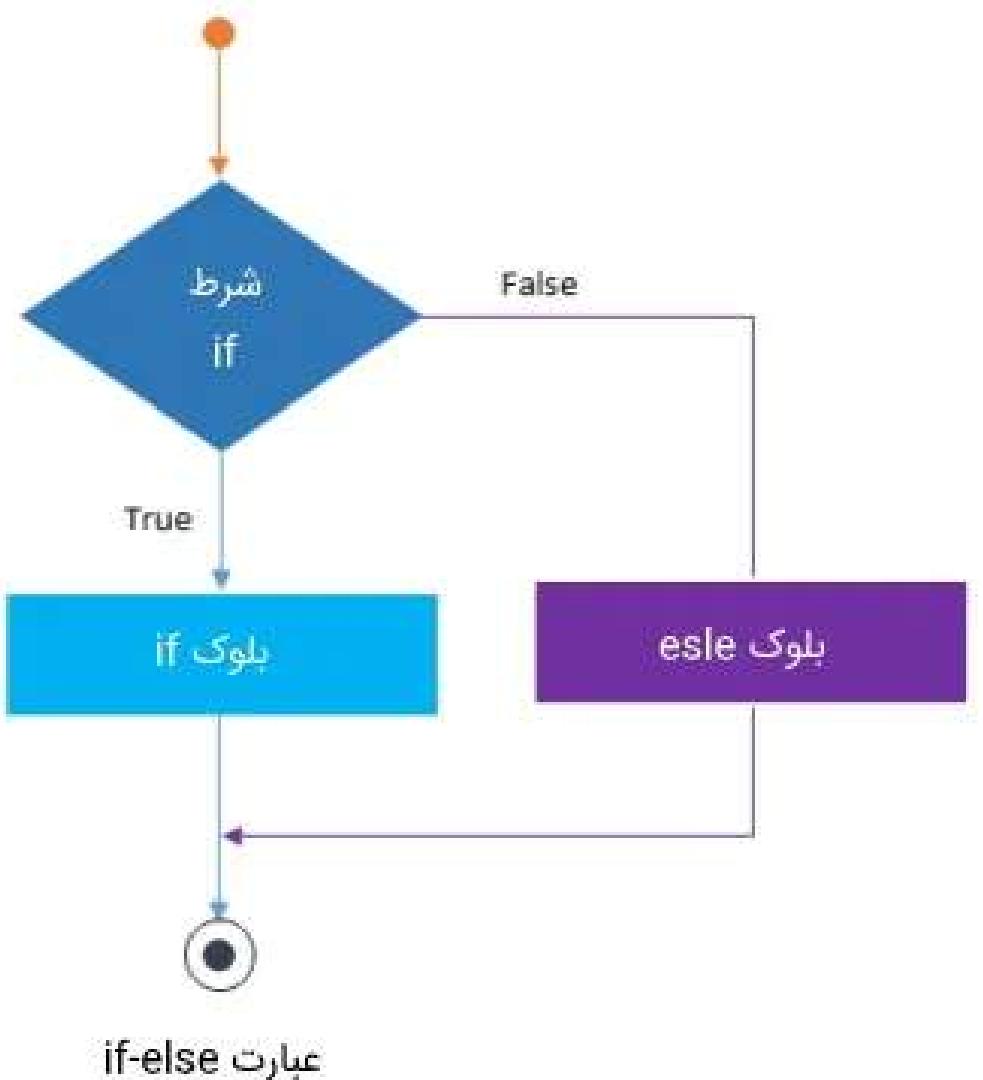
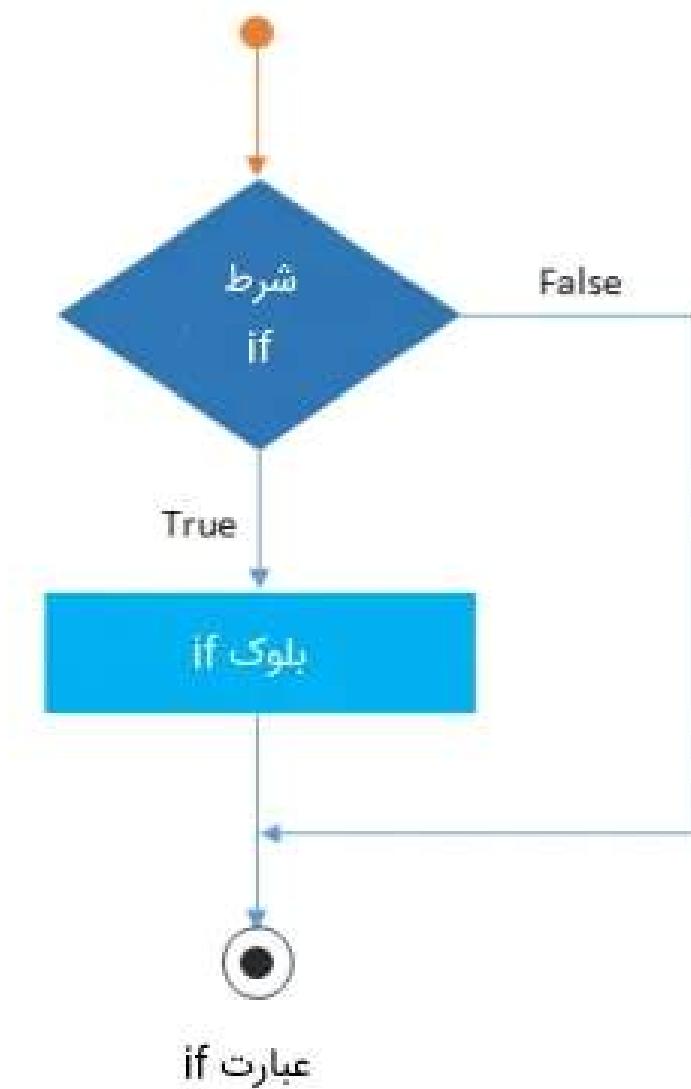
pop() Removes the element at the specified position

remove() Removes the item with the specified value

reverse() Reverses the order of the list

sort() Sorts the list

دستور IF , ELSE



ساختار IF , ELSE

کلمه

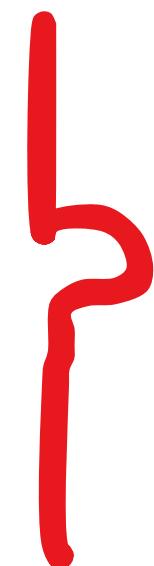
علامت دو نقطه: IF شرط

گزاره ای که اگر درست باشد، رعایت یک تب فاصله
اجرا میشود.

کلمه
else

علامت دو نقطه:

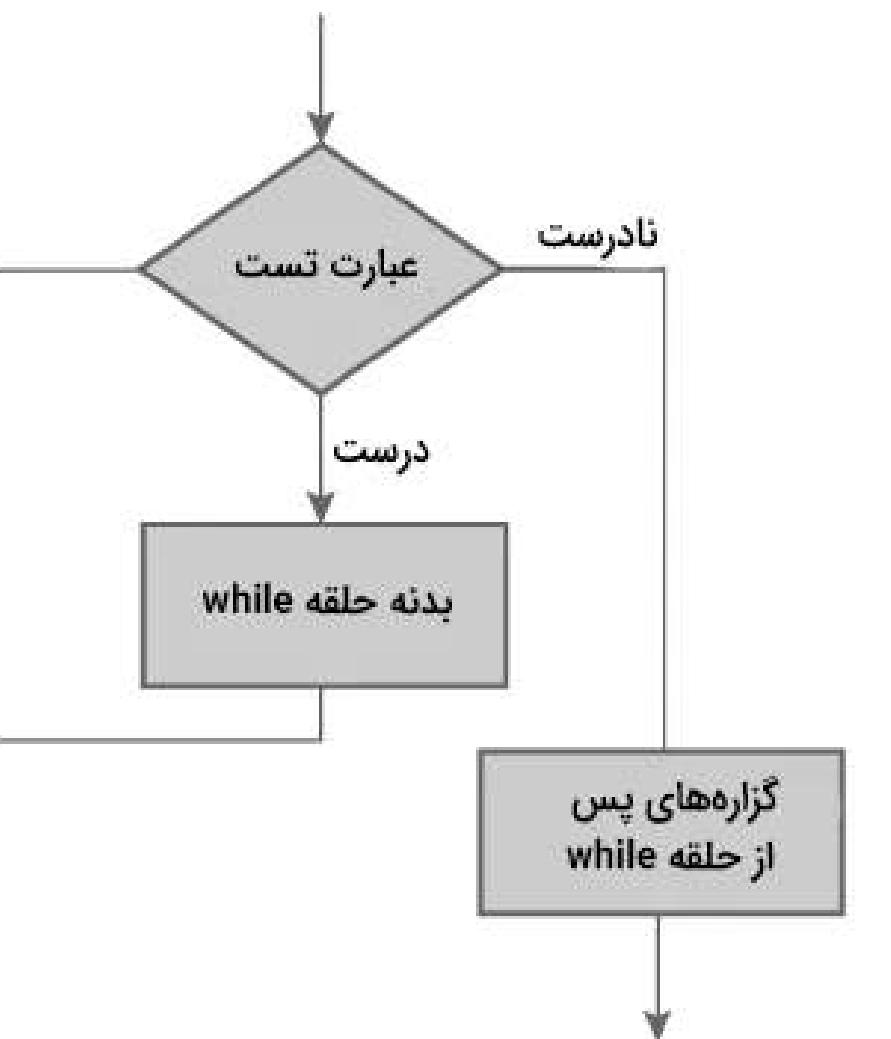
گزاره ای که اگر درست نباشد، رعایت یک تب فاصله
اجرا میشود.



اختیاری

نکته : همواره در شرط از عملگرهای مقایسه (== > < , > , ==) استفاده میشود.

حلقه ها for , while



ساختار for , while

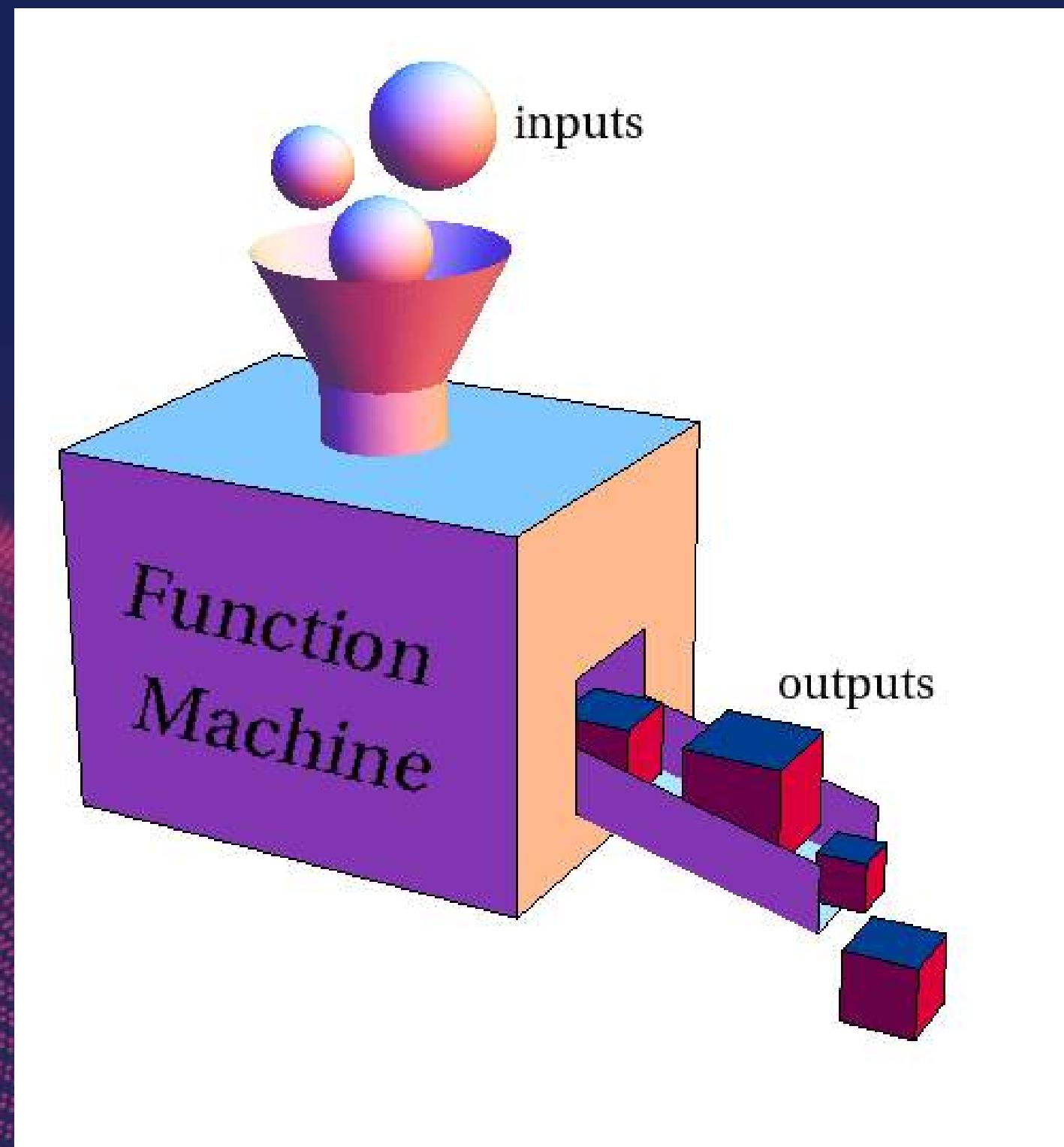
کلمه

علامت دو نقطه: شرط ادامه حلقه

گزاره ای که تا پایان حلقه تکرار میشود رعایت یک تب فاصله

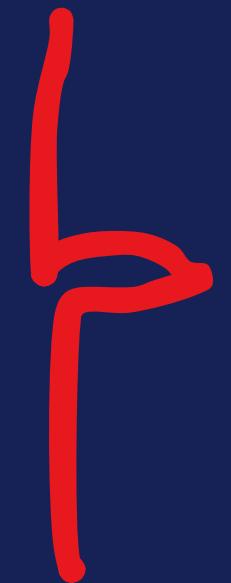
نکته: اکثرا در شرط ادامه حلقه از تابع داخلی range() استفاده میشود

تابع ها



ساختار تابع ها

تعریف تابع



فراخوانی
تابع



اسم تابع (آرگومان ها)

چیزی که بر میگرداند
(اختیاری)

کلمه دو نقطه:
(آرگومان ها) نام تابع

عملیات رعایت یک تب فاصله

کلمه def

Mdule

یا یک کتابخانه یعنی پر از هزاران تابع که میتوانیم فقط با صدا زدن تابع و دادن ورودی از آن خروجی بگیریم

طریقه وارد کردن :
ابتدا باید آن کتابخانه را دانلود کنیم :
`pip install numpy`

بعد آن را در کدمون وارد کنیم:

```
import numpy
```

میتوانیم اسم مخفف بزاریم

```
import numpy as np
```

یا یک تابع خاص رو بکشیم بیرون

```
from numpy import array
```

Import math

```
x = math.sqrt(64)
```

```
x = math.ceil(1.4)
y = math.floor(1.4)
math.cos()
math.sin()
math.exp()
math.log()
math.pow()
```

```
math.pi
math.e
math.tau
math.nan
math.inf
```

import statistics

```
statistics.mean()
statistics.median()
statistics.median_high()
statistics.median_low()
statistics.mode()
statistics.stdev()
statistics.pstdev()
statistics.variance()
statistics.pvariance()
```

Import random

<u>random()</u>	return float number between 0 and 1
<u>randrange()</u>	return int number between given range
<u>randint()</u>	return int number between given range
<u>uniform()</u>	return float number between given range uniform distribution
<u>gauss()</u>	return float number between given range gaussian distribution

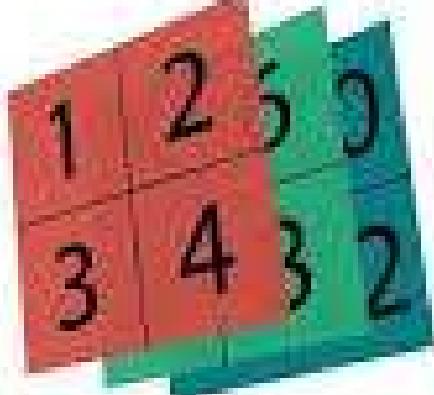
<u>shuffle()</u>	take a list and order that randomly
<u>choice()</u>	choice one number among one list
<u>choices()</u>	choice list among one list



کتابخانه نامپای

این کتابخانه برای محاسبات استفاده می‌شود و مانند یک لیست ما داریم اما لیستموں فقط یک بعدی نیست و میتوانه چند بعدی باشه (مثل ماتریس) و خب هم راحت‌تر هست و هم سرعت بیشتری دارد و کار با ماتریس‌ها برای ما راحت‌تر هست

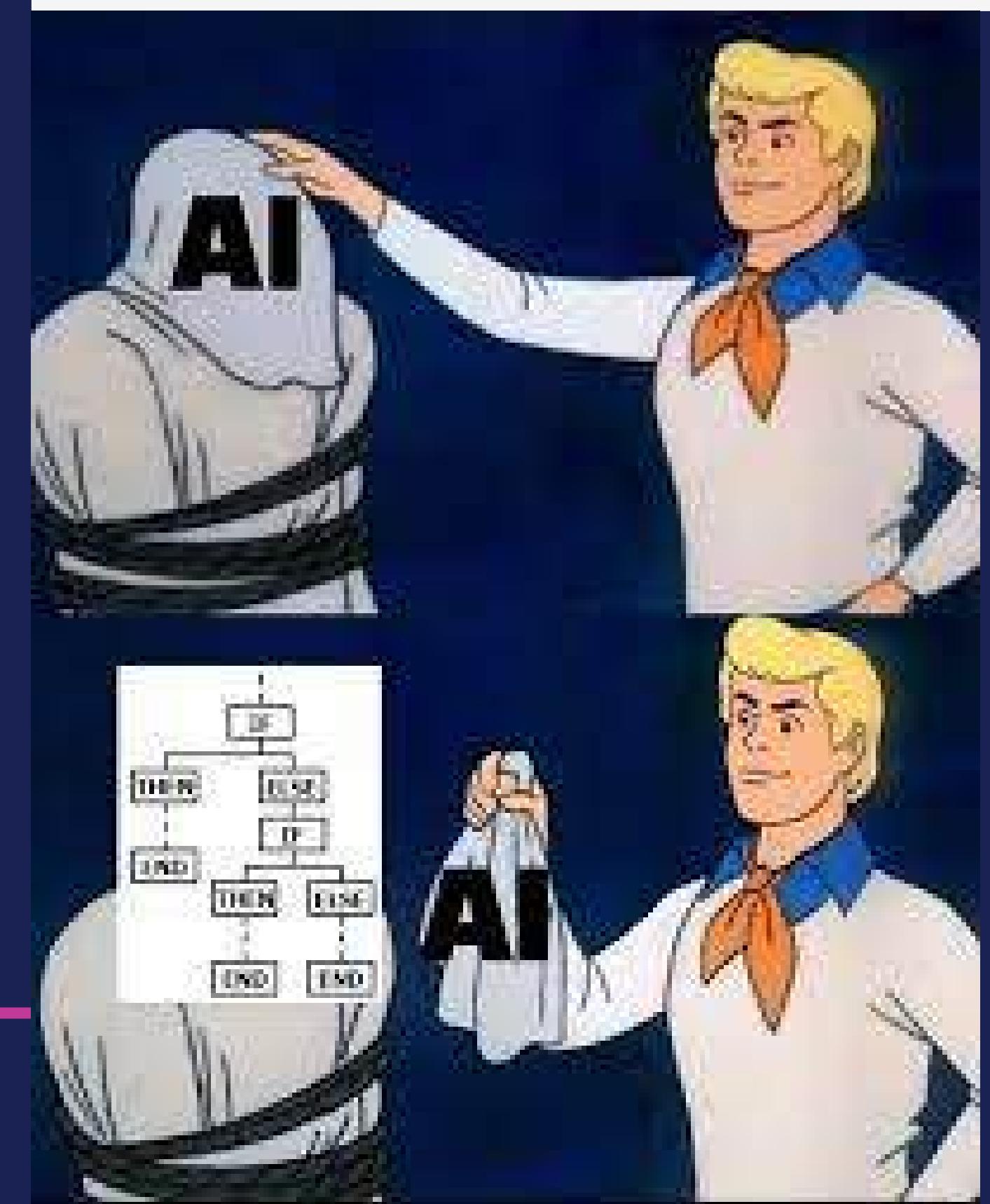
چند بعدی :
صفر بعدی: عدد

 Vector <code>np.array([1, 2])</code>	 Matrix <code>np.array([[1, 2], [3, 4]])</code>	 3D Matrix <code>np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]], [[9, 10], [11, 12]]])</code>
--	---	---

Section2: Machine learning

Ali Pilehvar Meibody
alipilehvar1999@gmail.com
ai.course22.alipilehvar@gmail.com

تقاوت برنامه و هوش مصنوعی

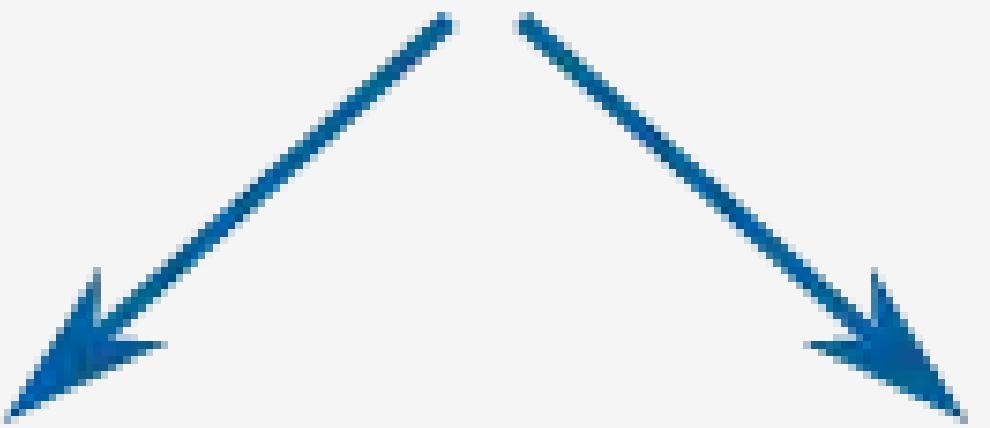


```
import pandas as pd  
data=pd.DataFrame([[0,200],[4,280],[8,310],[16,350],[24,400]],columns=[ 'X', 'Y'])
```

data - DataFrame

Index	X	Y
0	0	200
1	4	280
2	8	310
3	16	350
4	24	400

Data

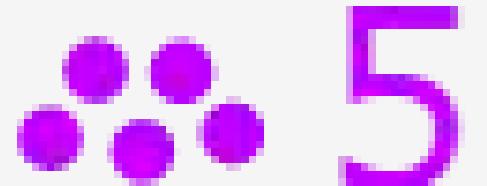


Qualitative

"It was great fun"

Quantitative

Discrete

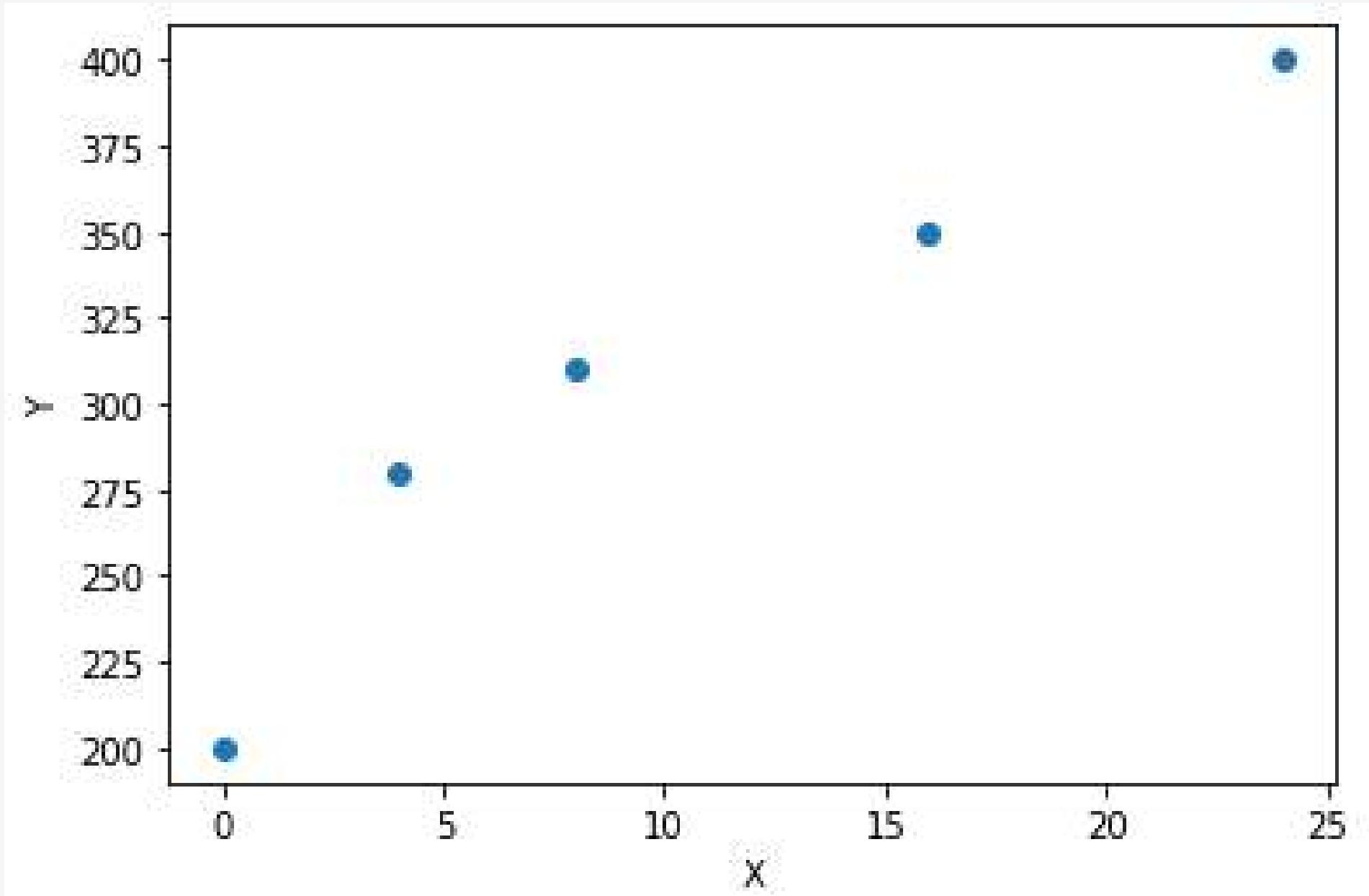


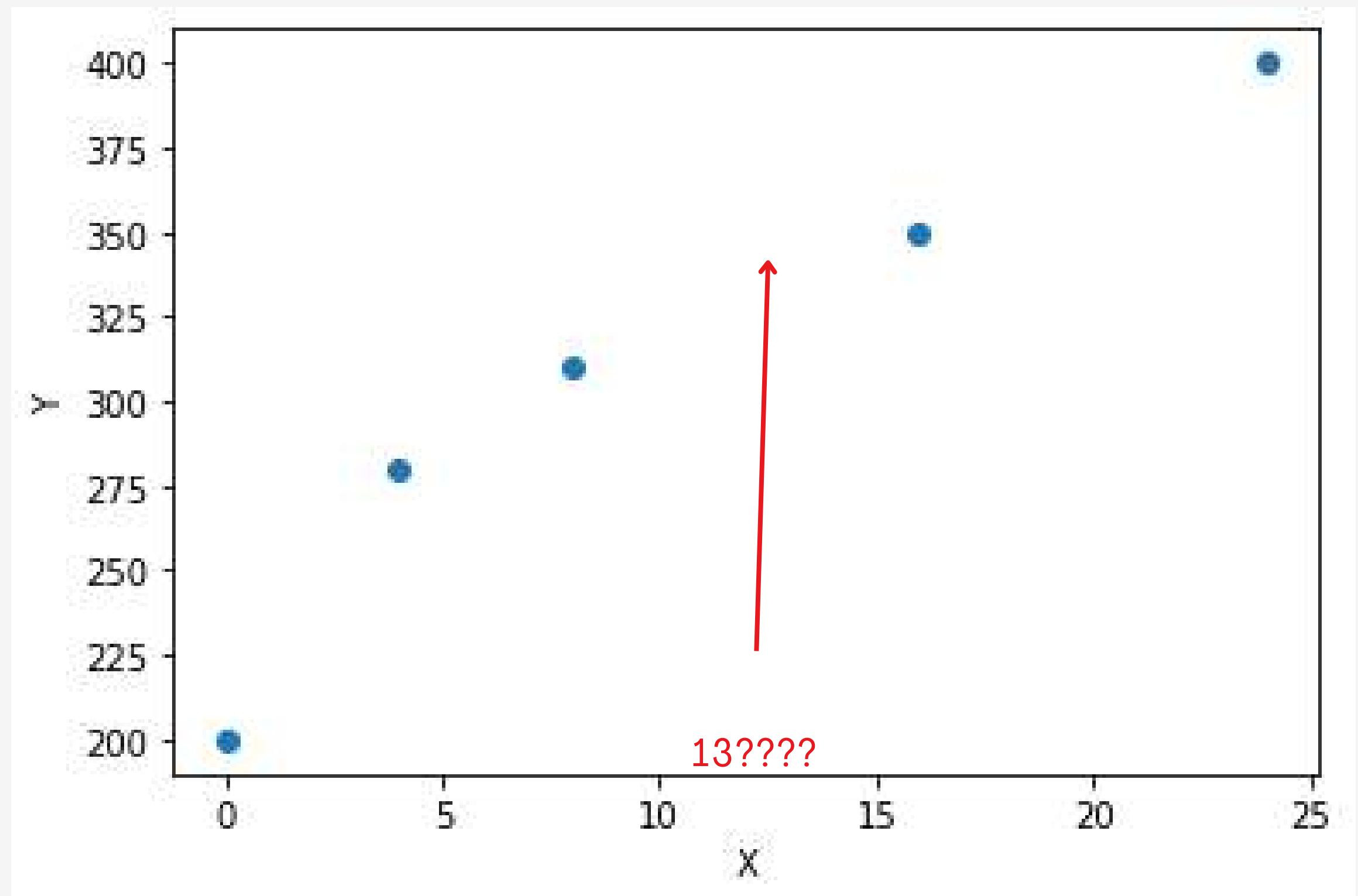
Continuous

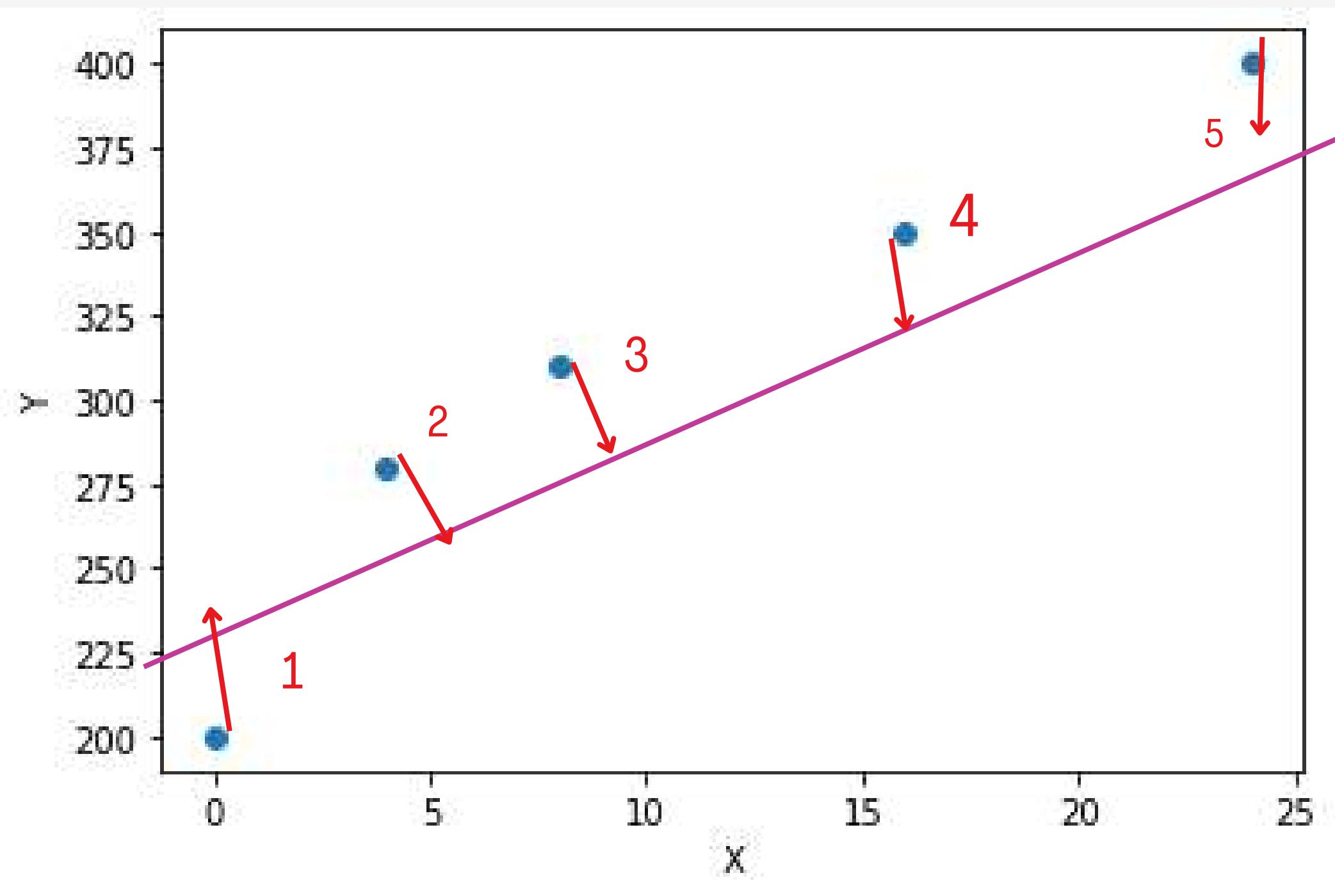
3.265...



```
import matplotlib.pyplot as plt  
plt.scatter(data[ 'X' ],data[ 'Y' ])
```





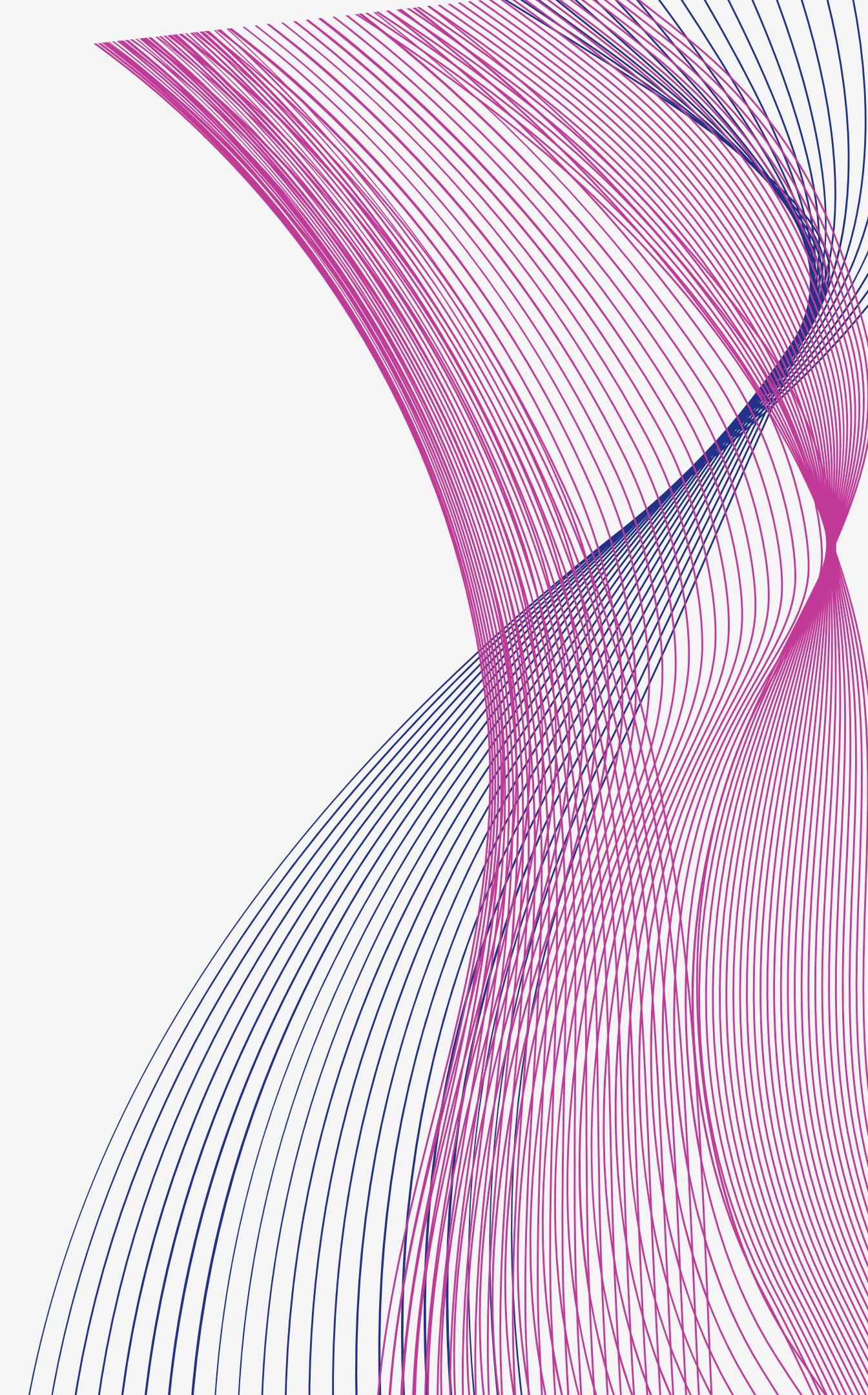


ALL DISTANCES=



COST FUNCTION:

$$((2) + (2) + (2) + (2) + (2)) \times 100 = 1000$$



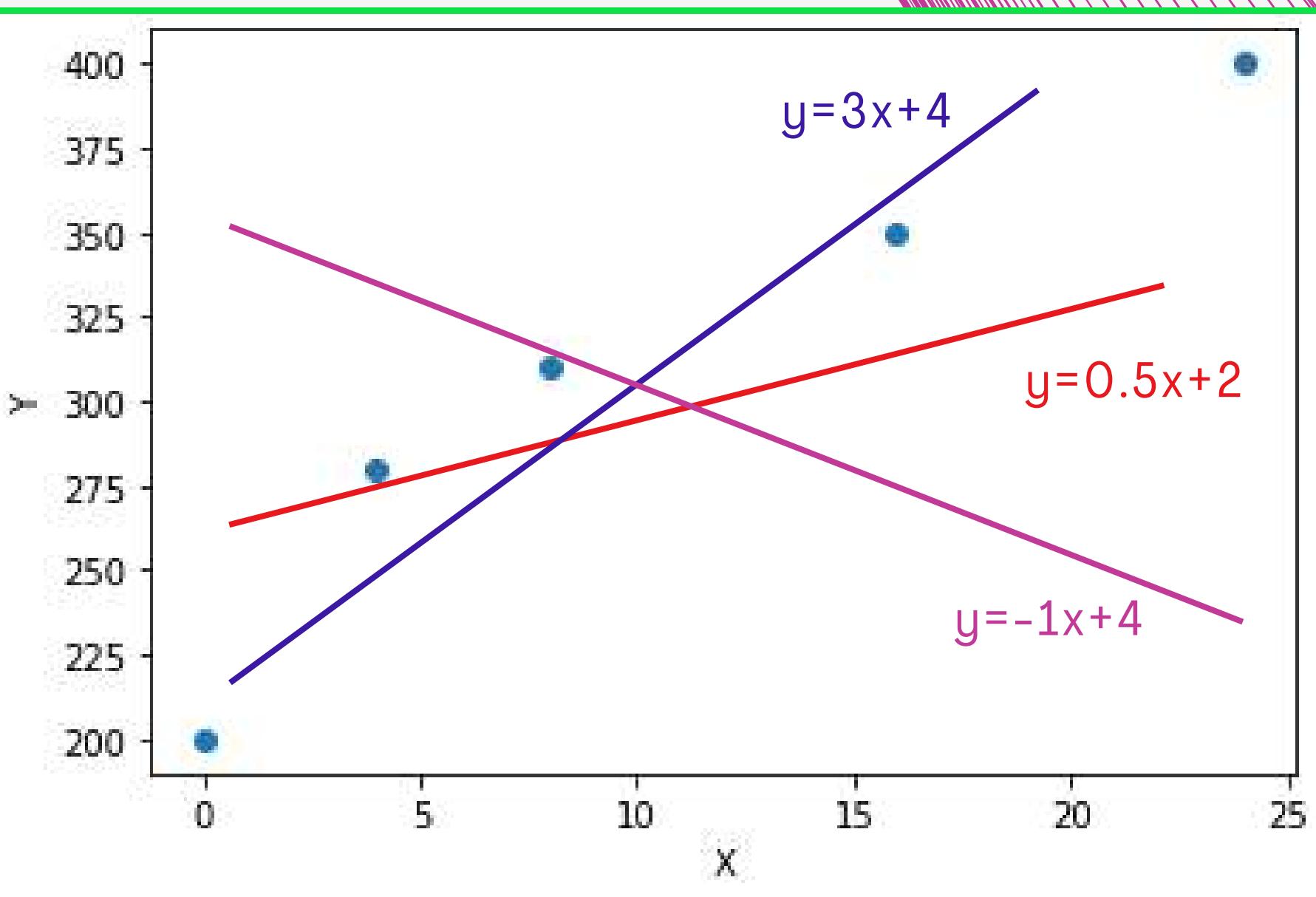
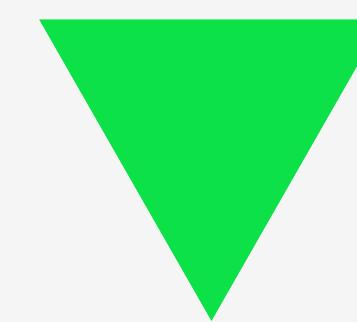
optimization

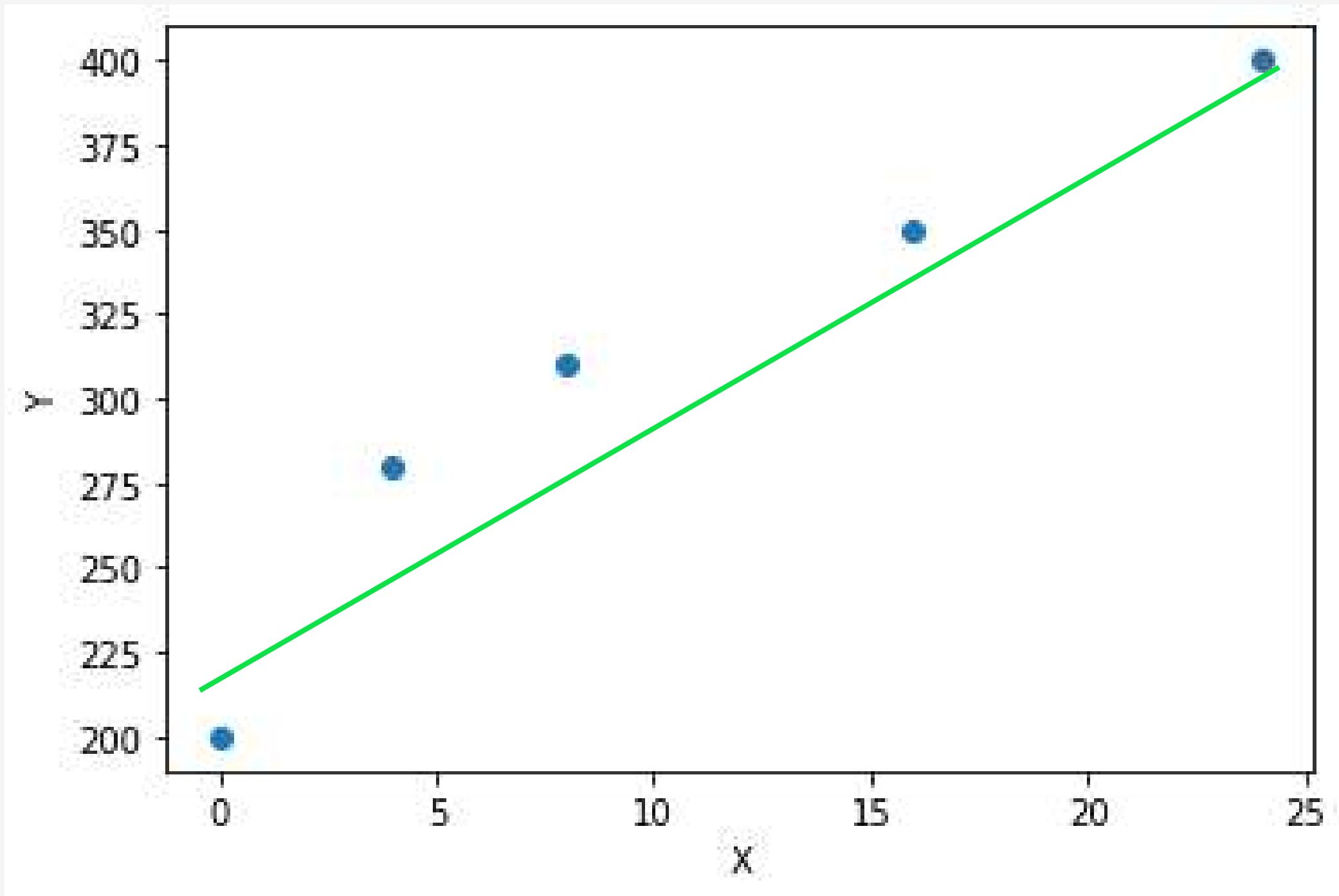
COST FUNCTION1

COST FUNCTION2

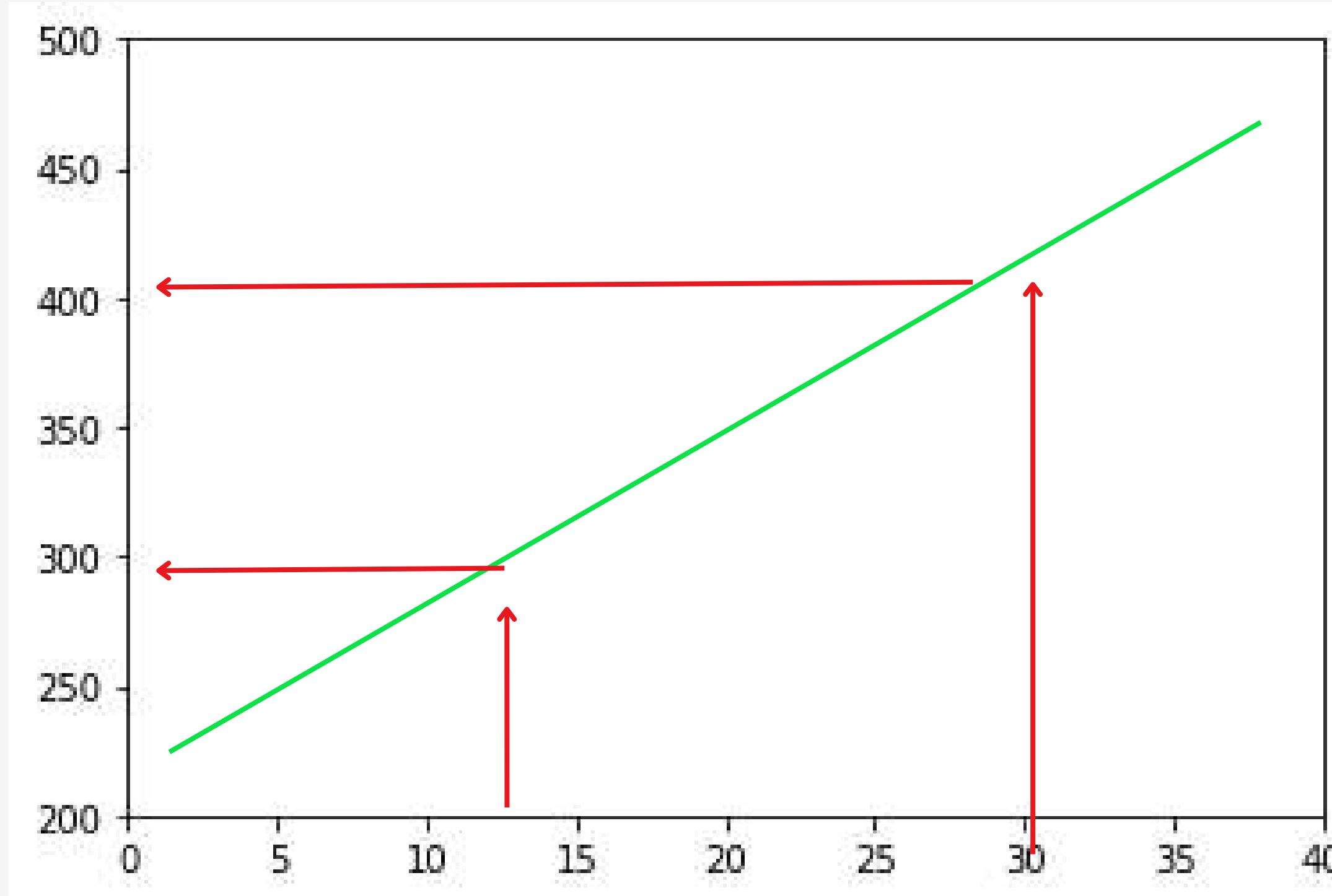
COST FUNCTION3

.....





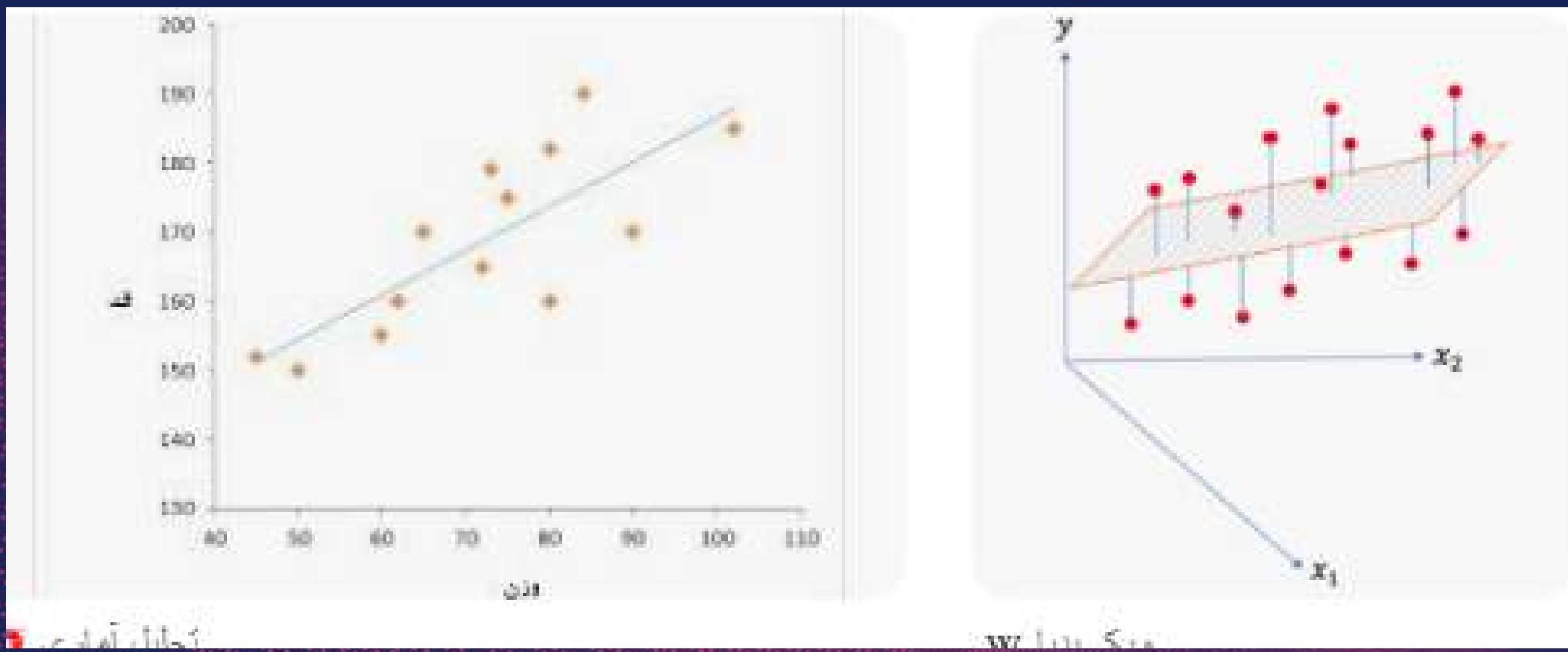
$$Y = Ax + b$$



$$Y = Ax + b$$

همچنین میشه برای دو بعدی ها هم این کار را انجام داد

$$Y = AX + BX + \hat{C}$$



$$Y = AX_1 + BX_2 + CX_3 + DX_4 + \dots + Z$$

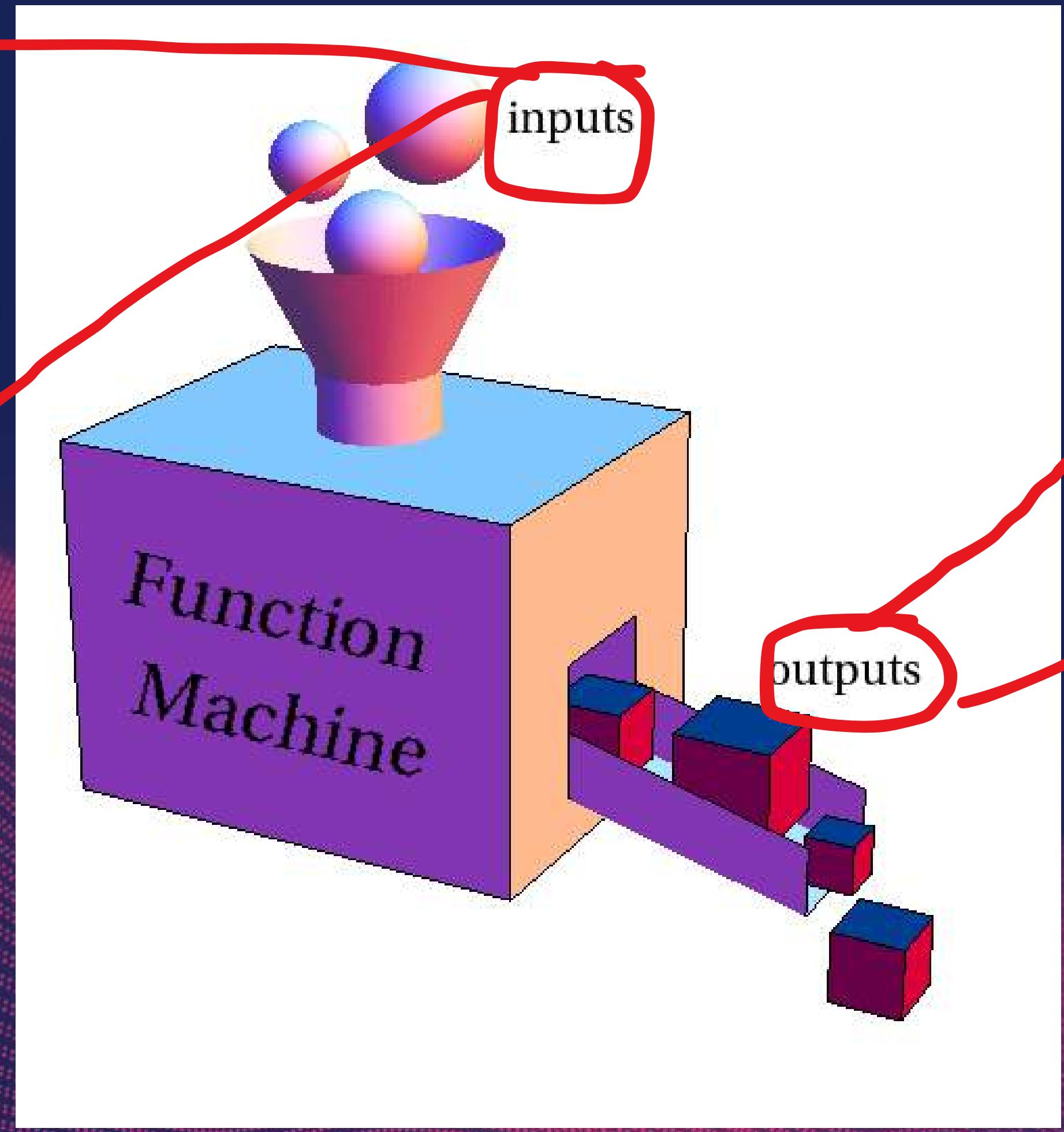
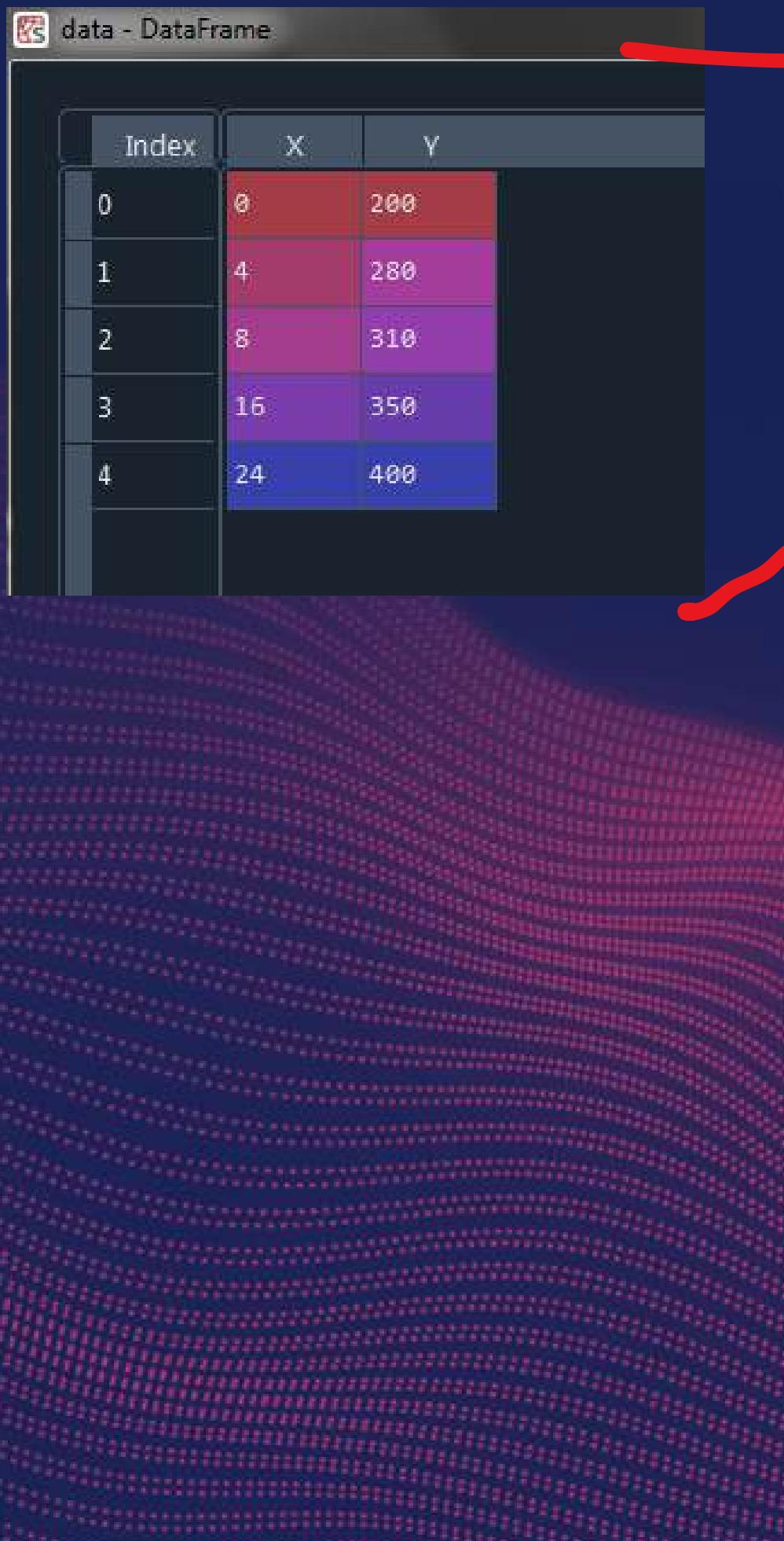
یادگیری ماشین دو مرحله دارد

1- مرحله یادگیری training

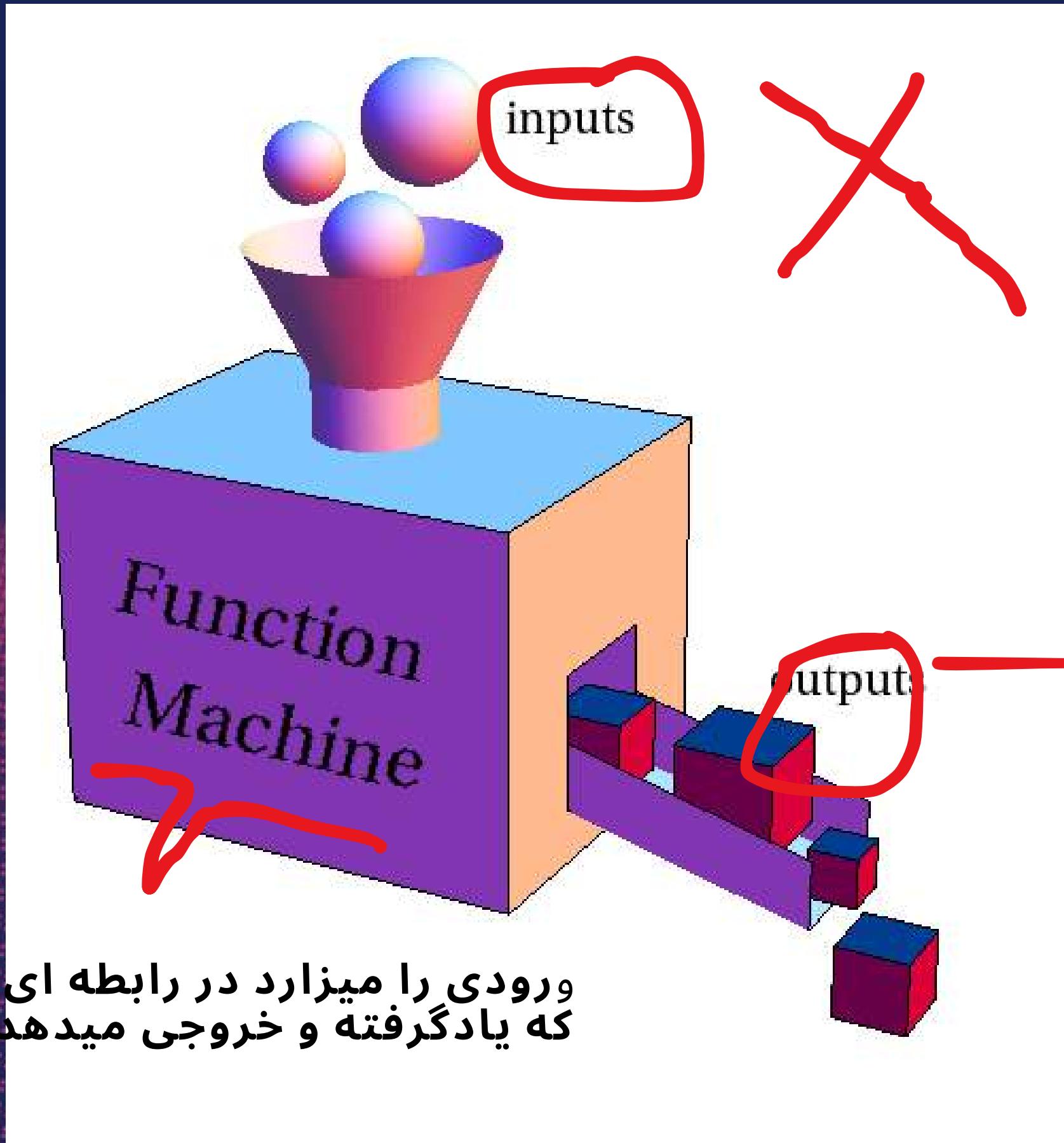
2- مرحله پیش بینی prediction

مرحله یادگیری

$$Y = Ax + b$$

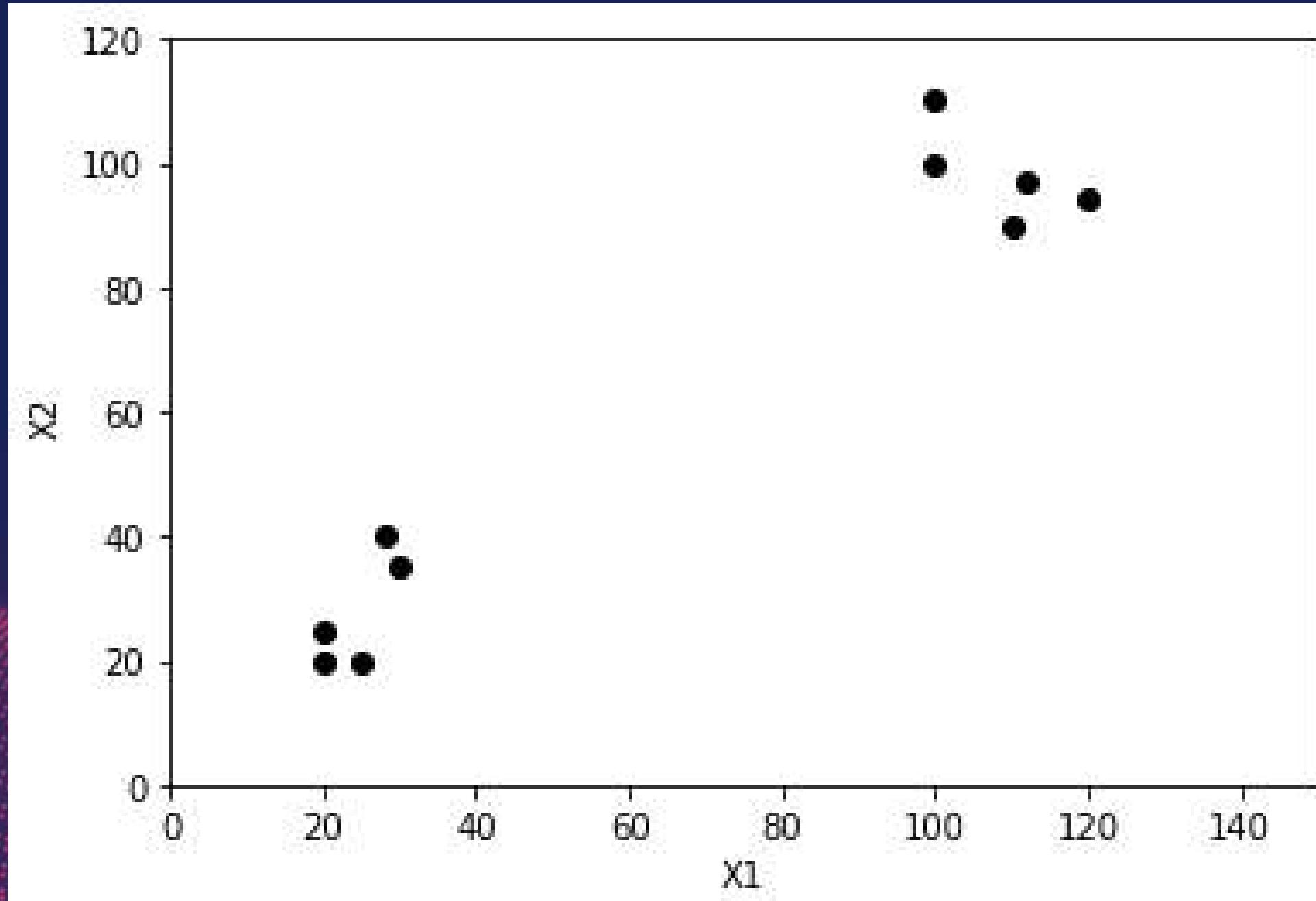


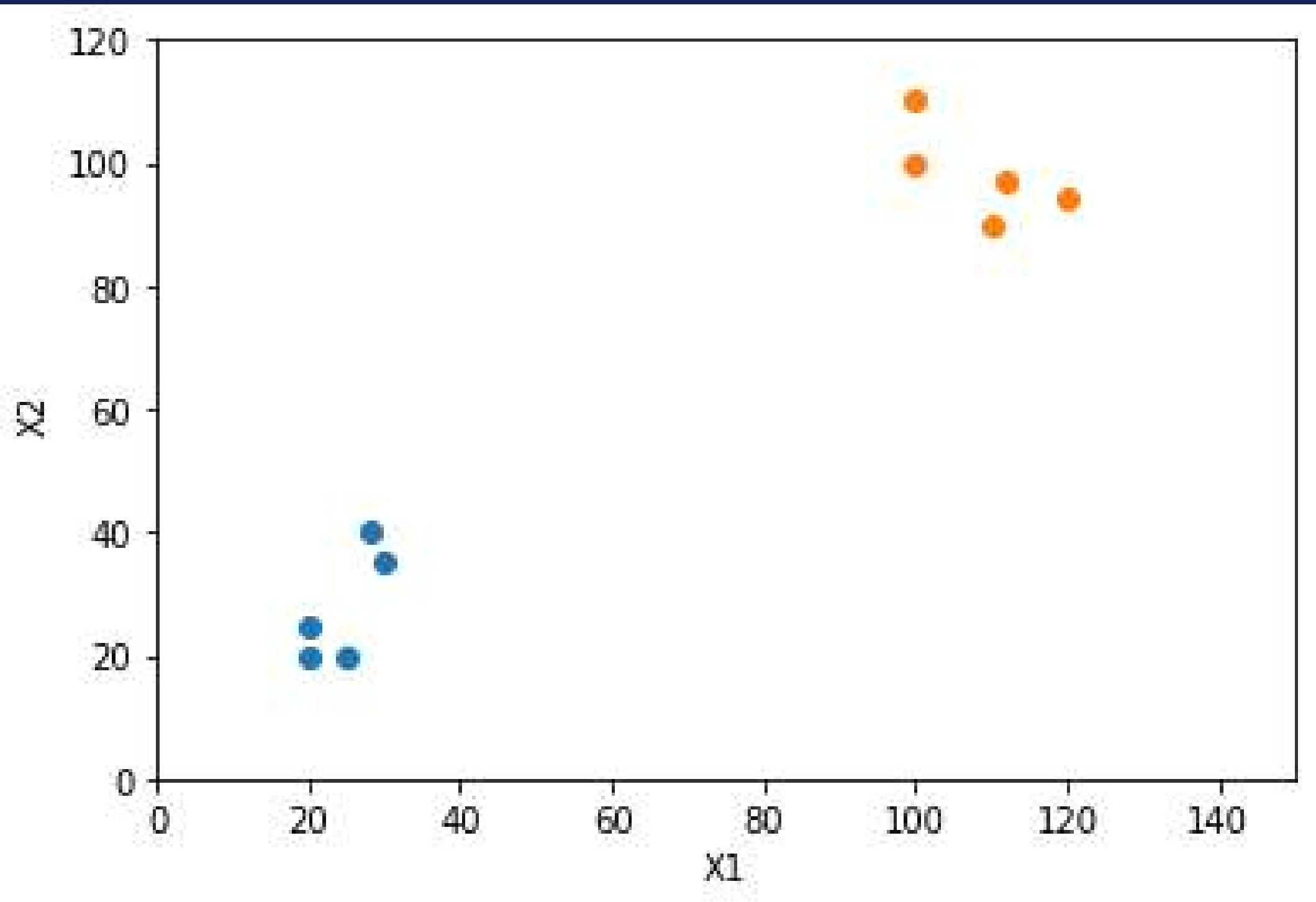
مرحله پیش بینی

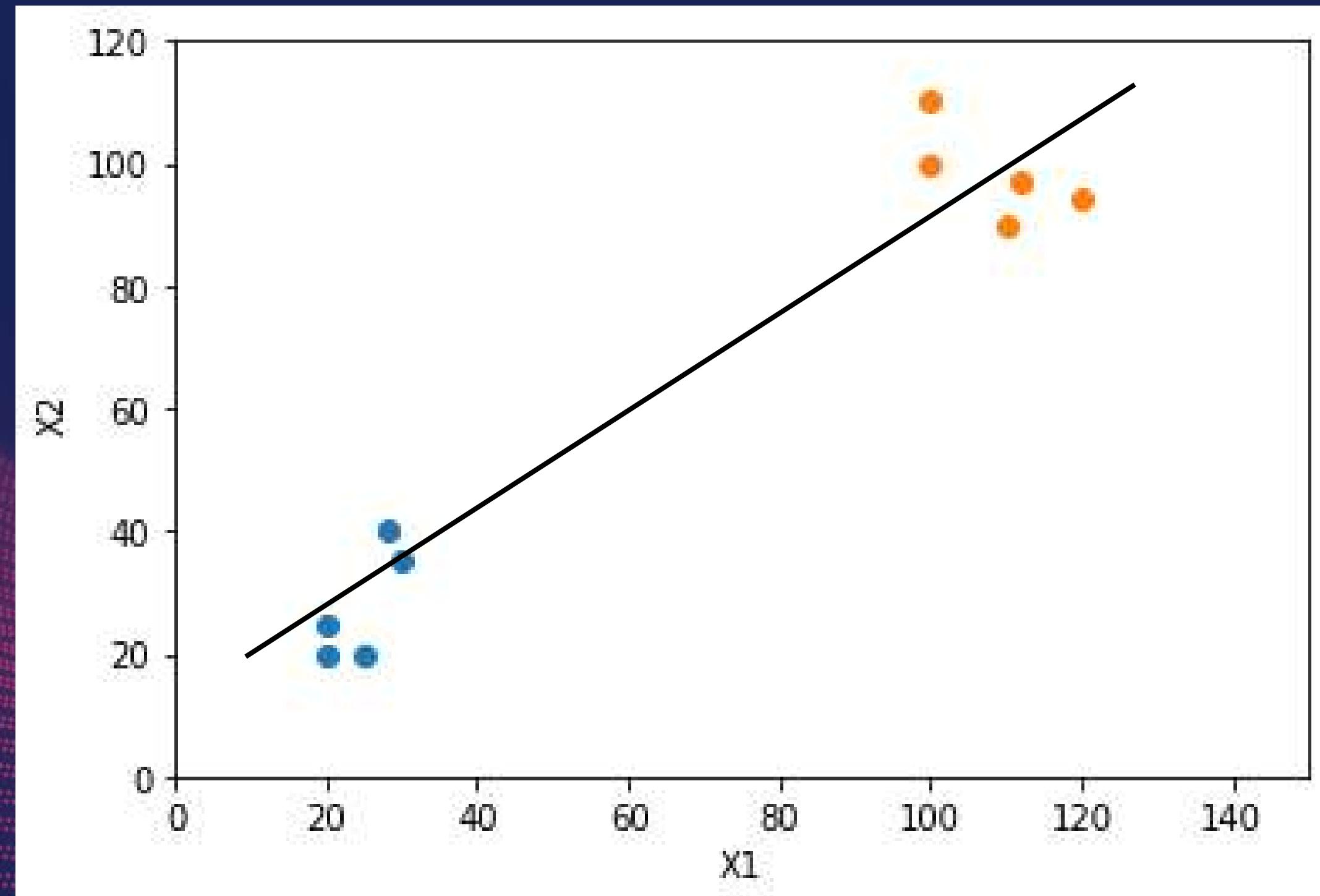


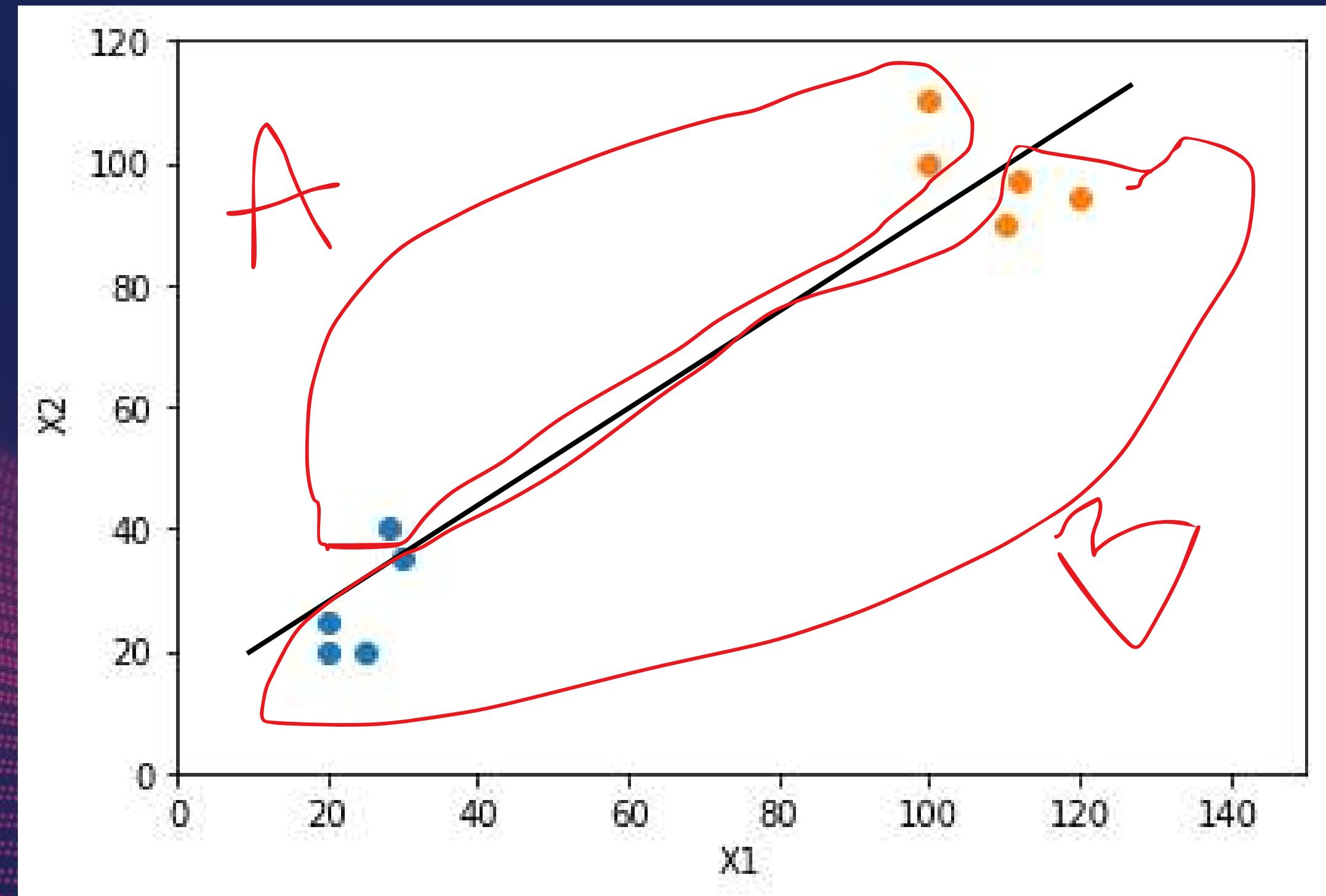
دسته بندی classification

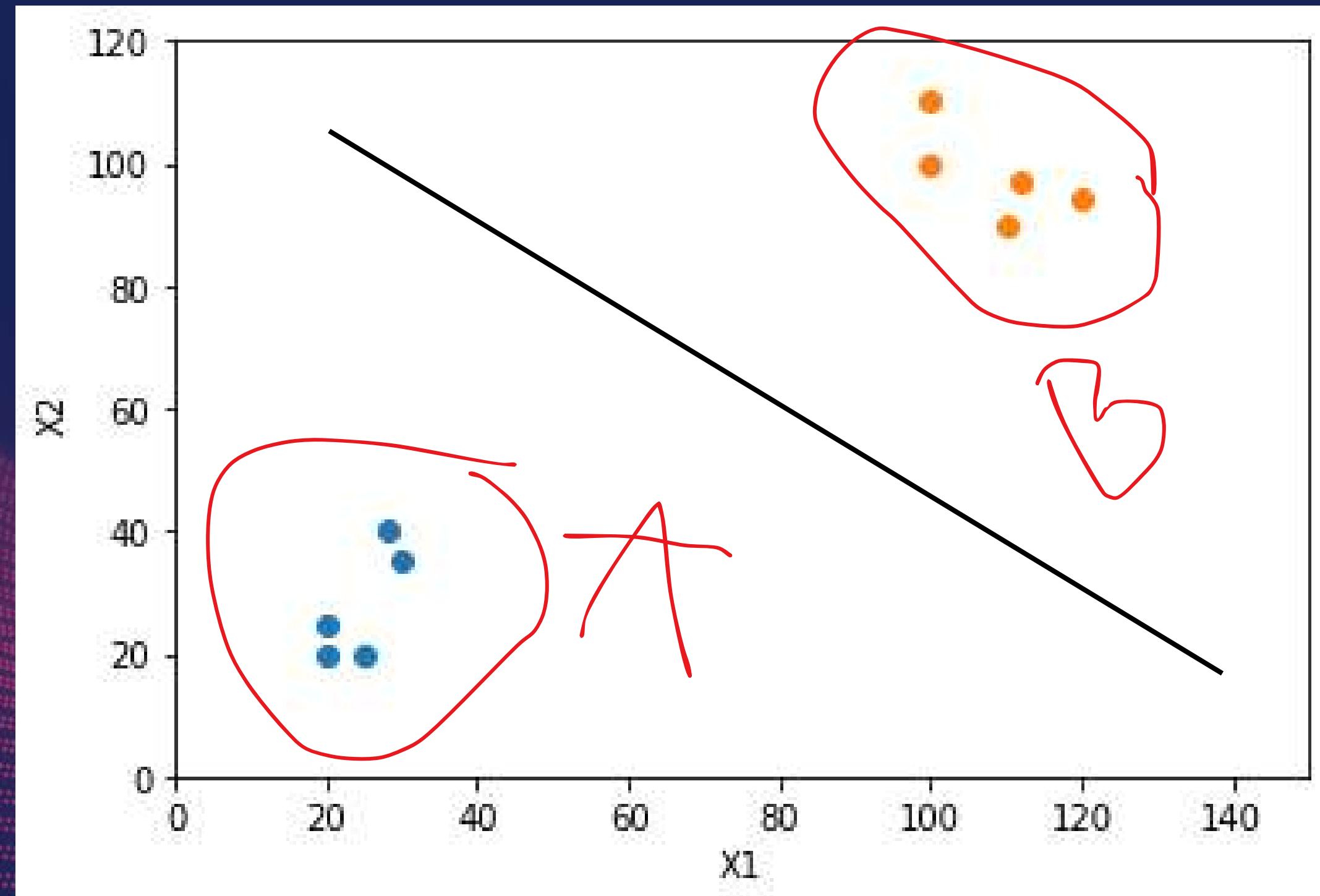
Index	X	Y
0	20	20
1	20	25
2	25	20
3	30	35
4	28	40
5	100	100
6	100	110
7	110	90
8	120	94
9	112	97



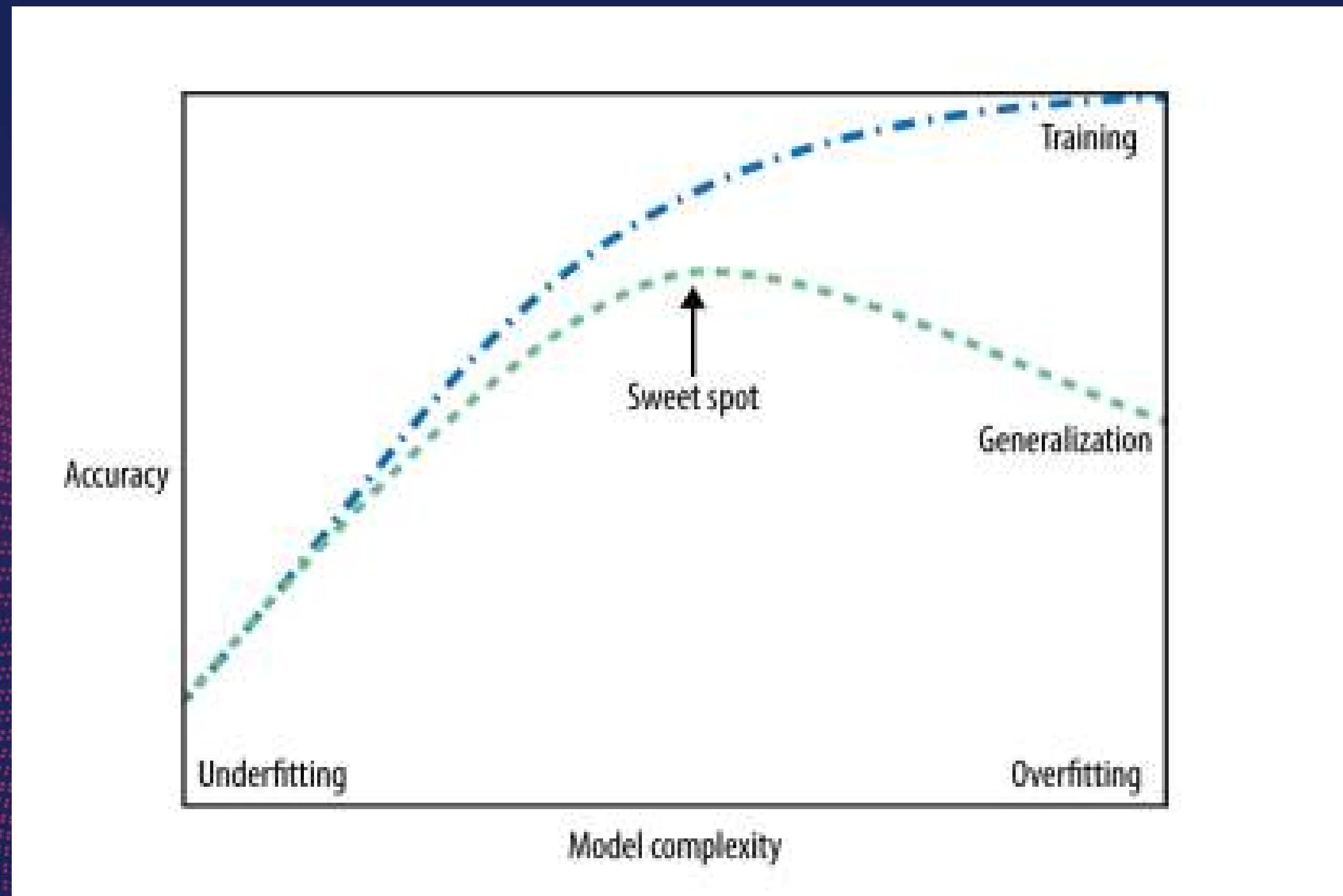




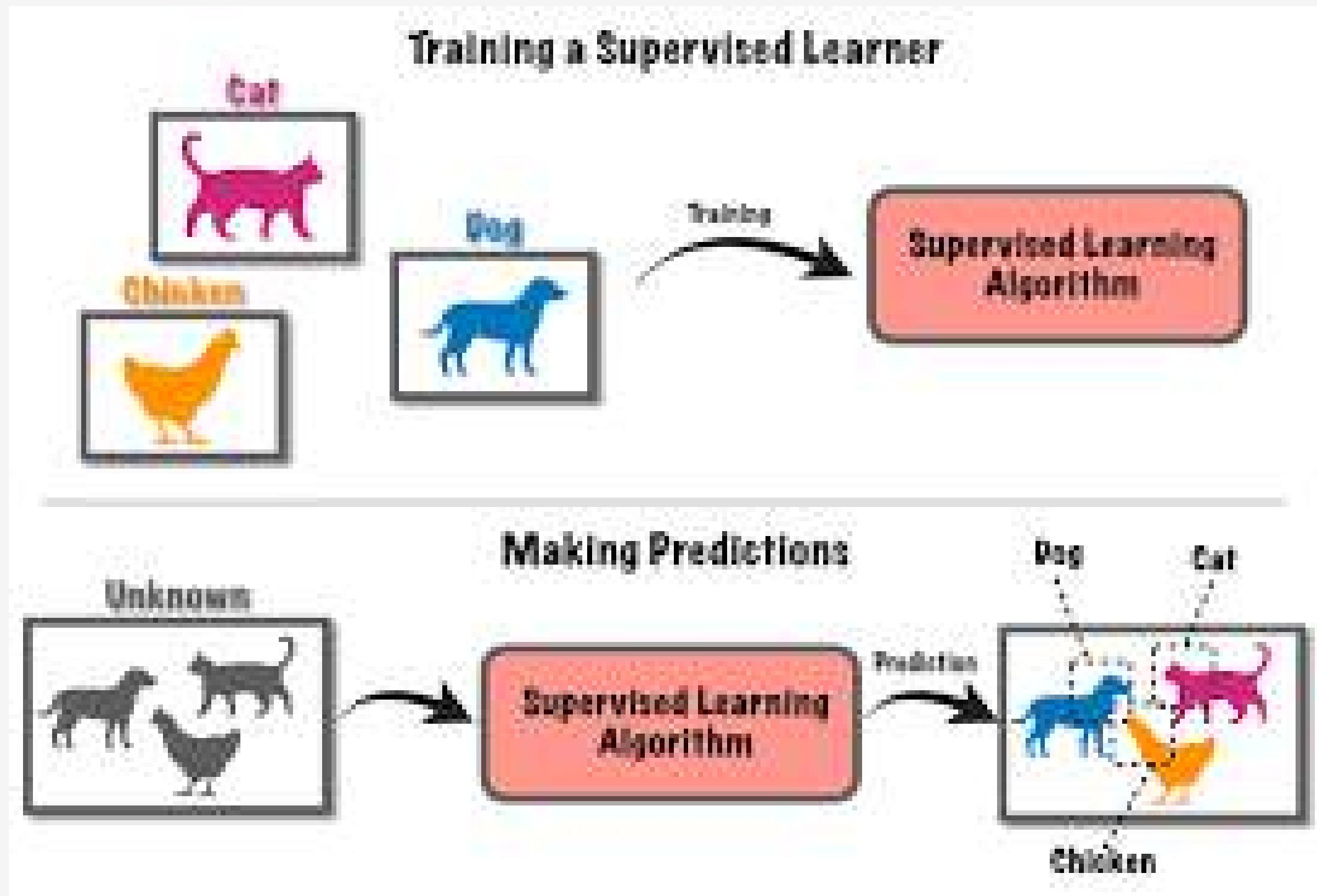




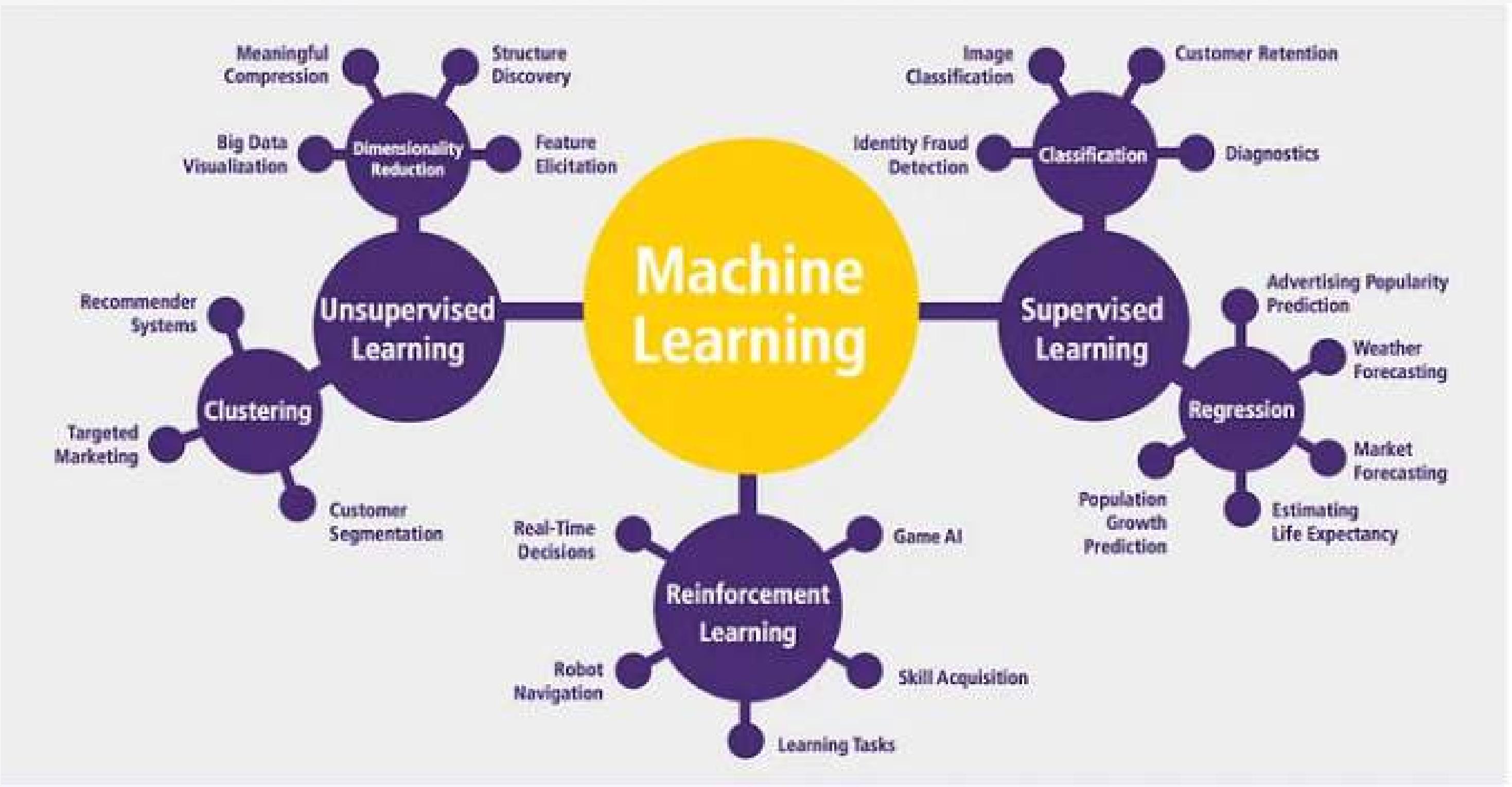
فرق هوش مصنوعی و رگرسیون ساده



مثال سگ و گربه برای درک بهتر



روش های یادگیری ماشین

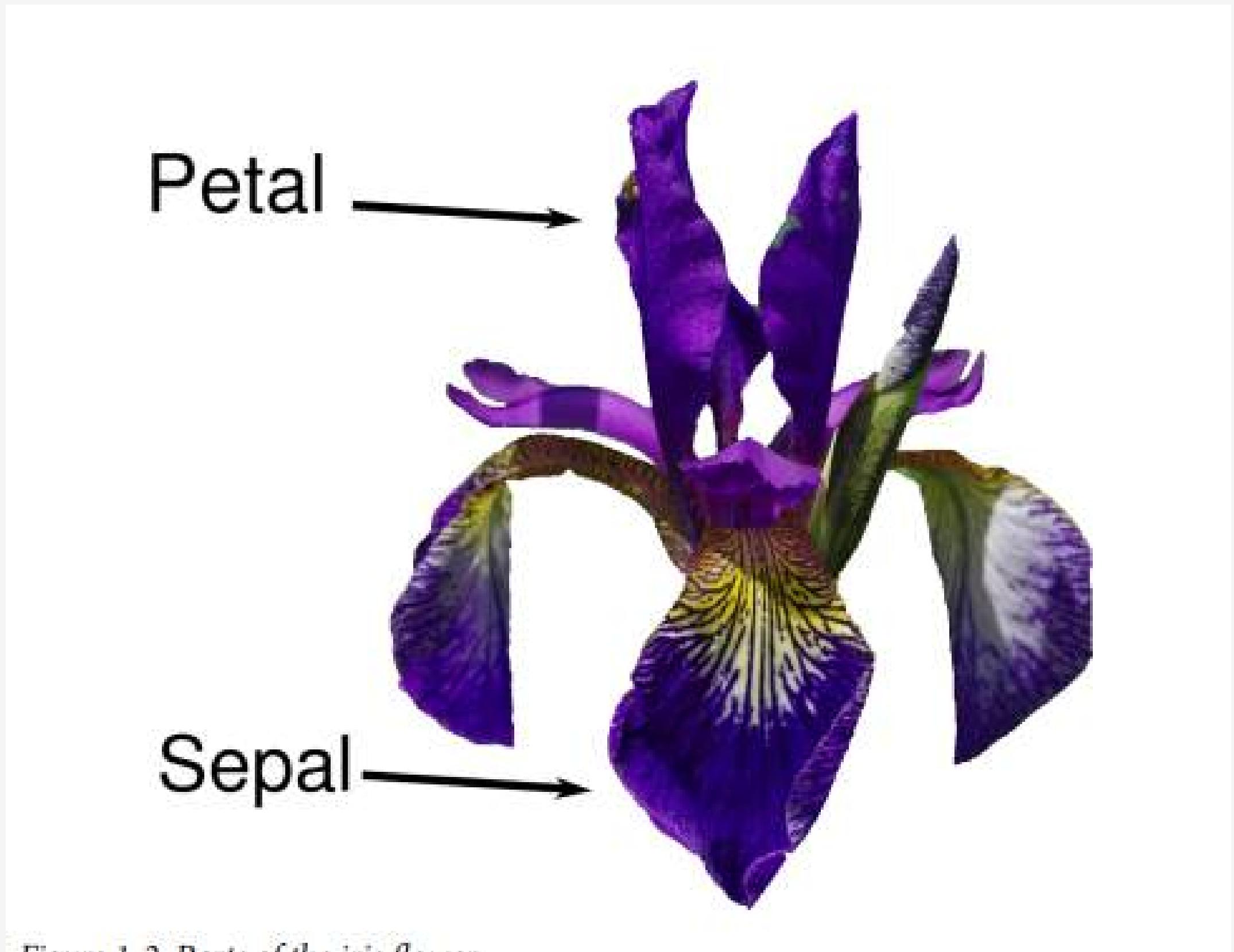


کتابخانه scikit learn



تمام ریاضیات و مدل های پیچیده آماری قابل استفاده در هوش مصنوعی و یادگیری ماشین را میتوانیم فقط با صدا زدن آن تابع در این کتابخانه بیابیم.

داده های گل آیریس

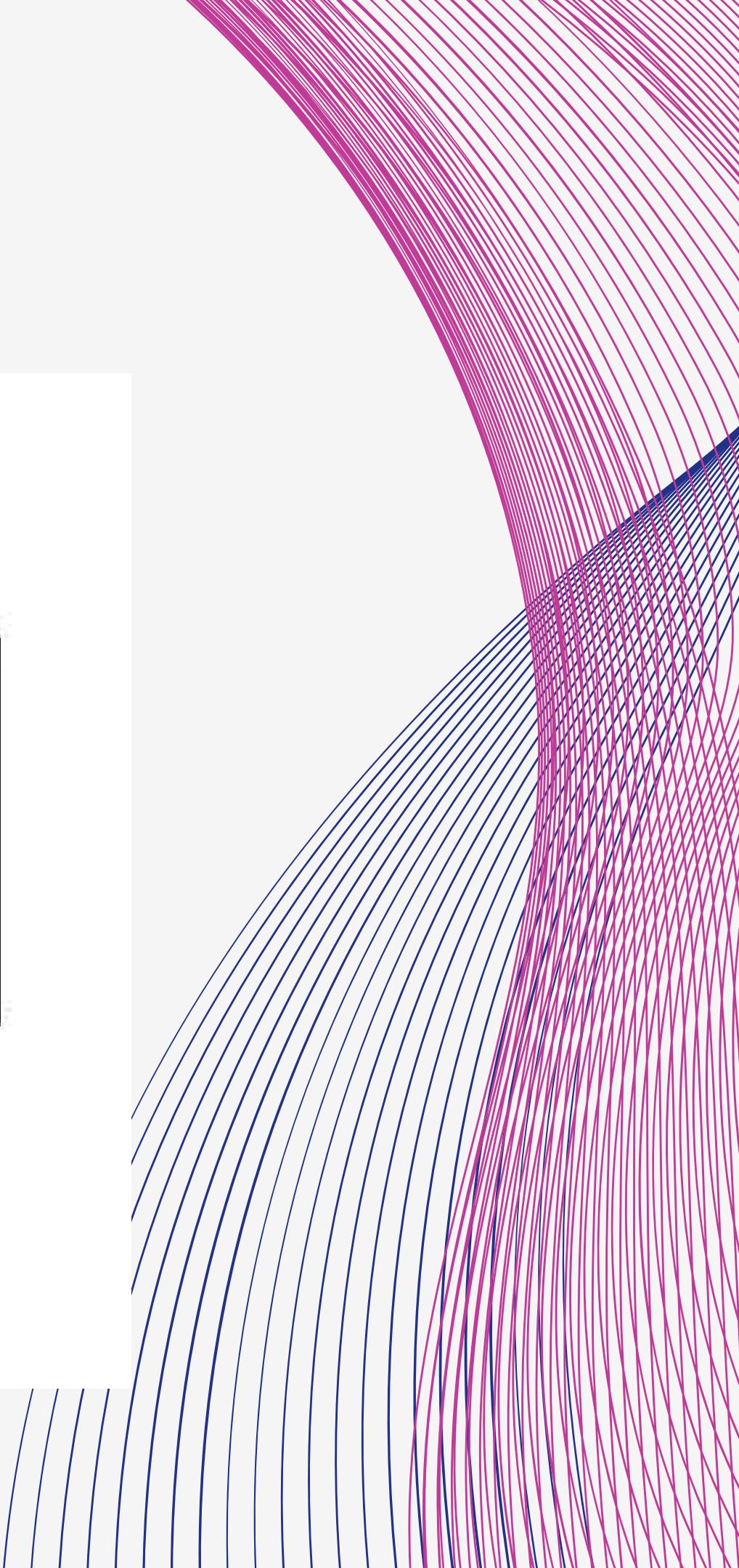
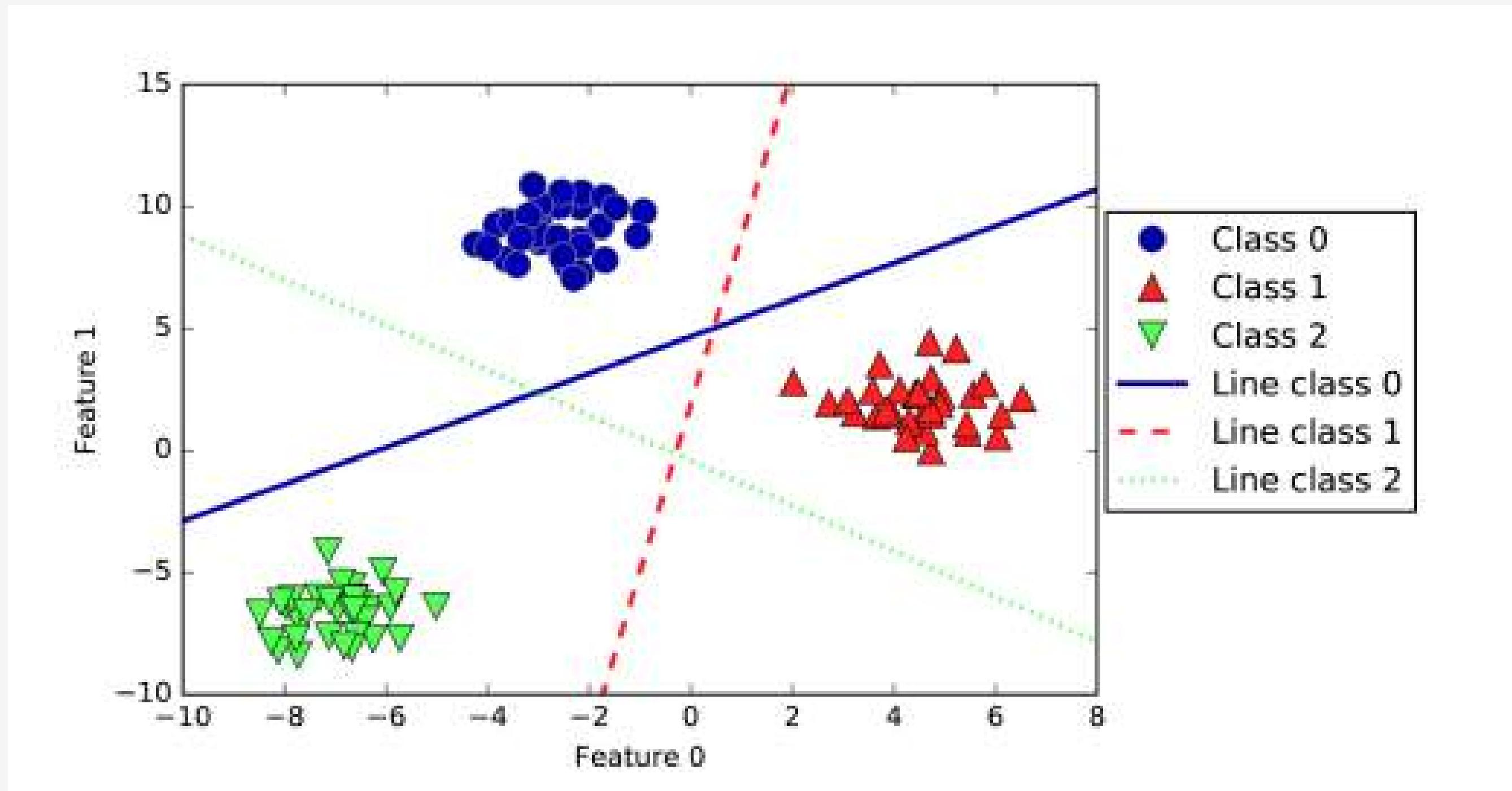


Flowers in the dataset

Logistic regression



for more 2 classes



SCORE - METRIC - ERROR

Scoring Classification	Function	Comment
'accuracy'	<code>metrics.accuracy_score</code>	
'balanced_accuracy'	<code>metrics.balanced_accuracy_score</code>	
'top_k_accuracy'	<code>metrics.top_k_accuracy_score</code>	
'average_precision'	<code>metrics.average_precision_score</code>	
'neg_brier_score'	<code>metrics.brier_score_loss</code>	
'f1'	<code>metrics.f1_score</code>	for binary targets
'f1_micro'	<code>metrics.f1_score</code>	micro-averaged
'f1_macro'	<code>metrics.f1_score</code>	macro-averaged
'f1_weighted'	<code>metrics.f1_score</code>	weighted average
'f1_samples'	<code>metrics.f1_score</code>	by multilabel sample
'neg_log_loss'	<code>metrics.log_loss</code>	requires <code>predict_proba</code> support
'precision' etc.	<code>metrics.precision_score</code>	suffixes apply as with 'f1'
'recall' etc.	<code>metrics.recall_score</code>	suffixes apply as with 'f1'
'jaccard' etc.	<code>metrics.jaccard_score</code>	suffixes apply as with 'f1'
'roc_auc'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovr'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovo'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovr_weighted'	<code>metrics.roc_auc_score</code>	
'roc_auc_ovo_weighted'	<code>metrics.roc_auc_score</code>	

```
from sklearn.metrics import confusion_matrix  
confusion = confusion_matrix(y_test, pred_logreg)
```

	TN	FP
negative class		
positive class	FN	TP
predicted negative		predicted positive

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

```
from sklearn.metrics import f1_score  
(f1_score(y_test, pred_tree))
```

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

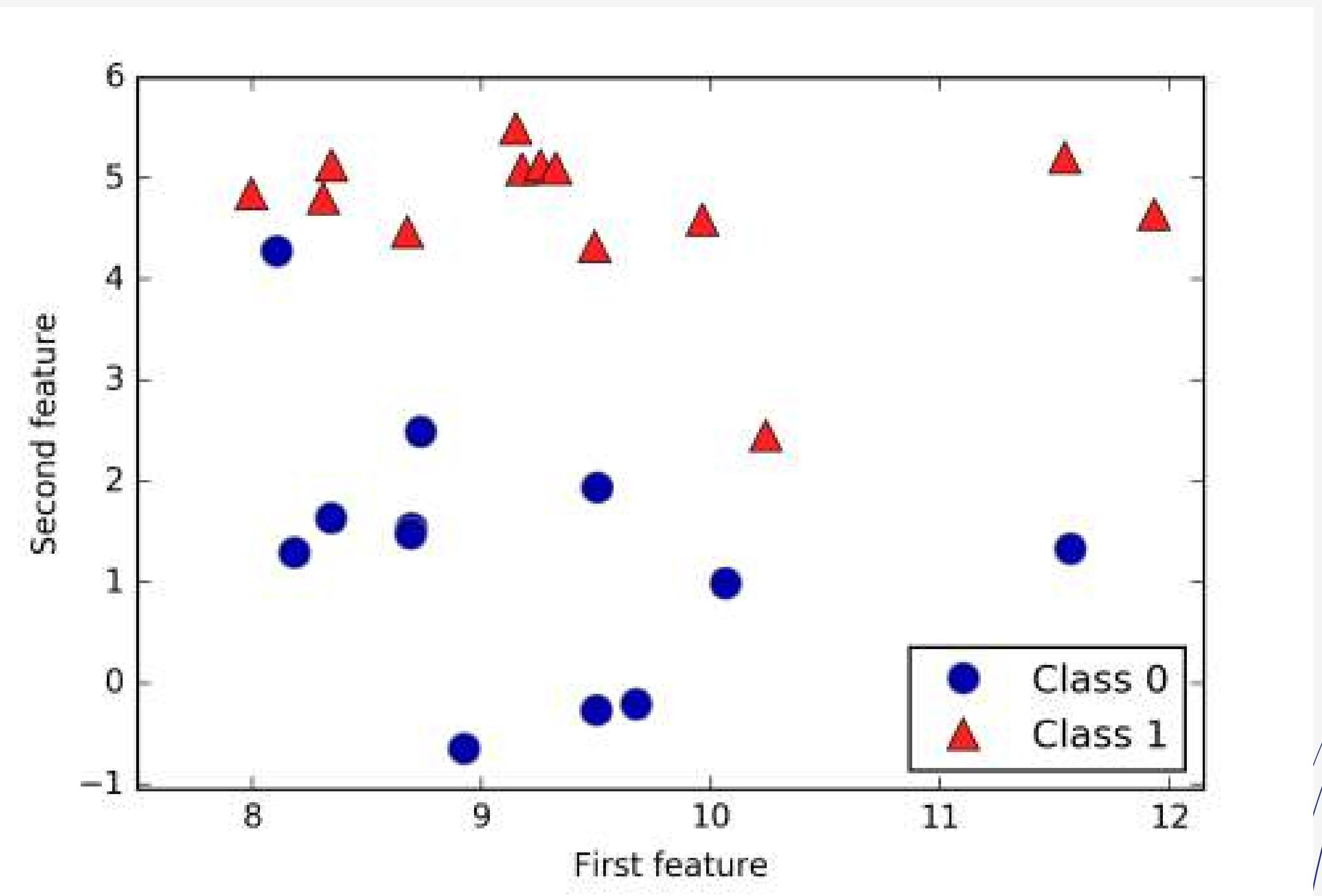
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

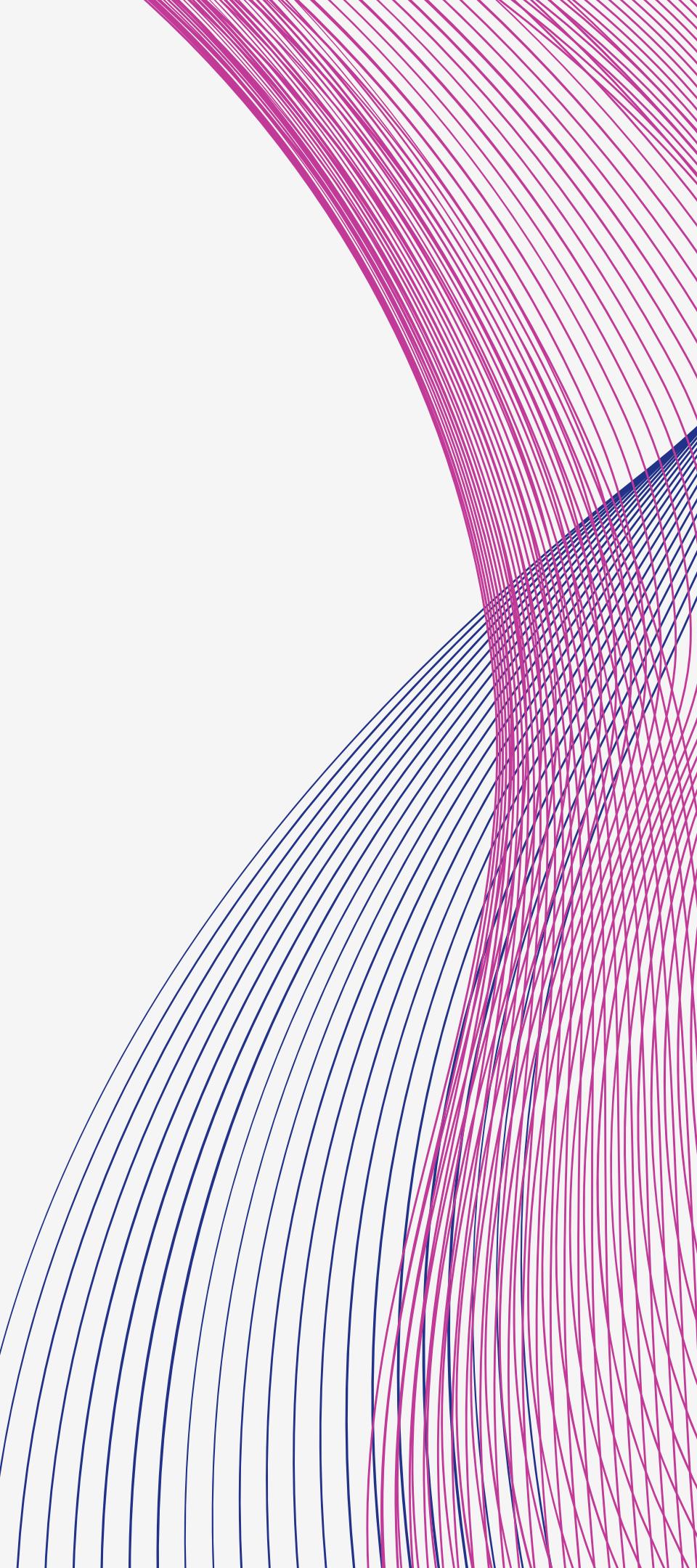
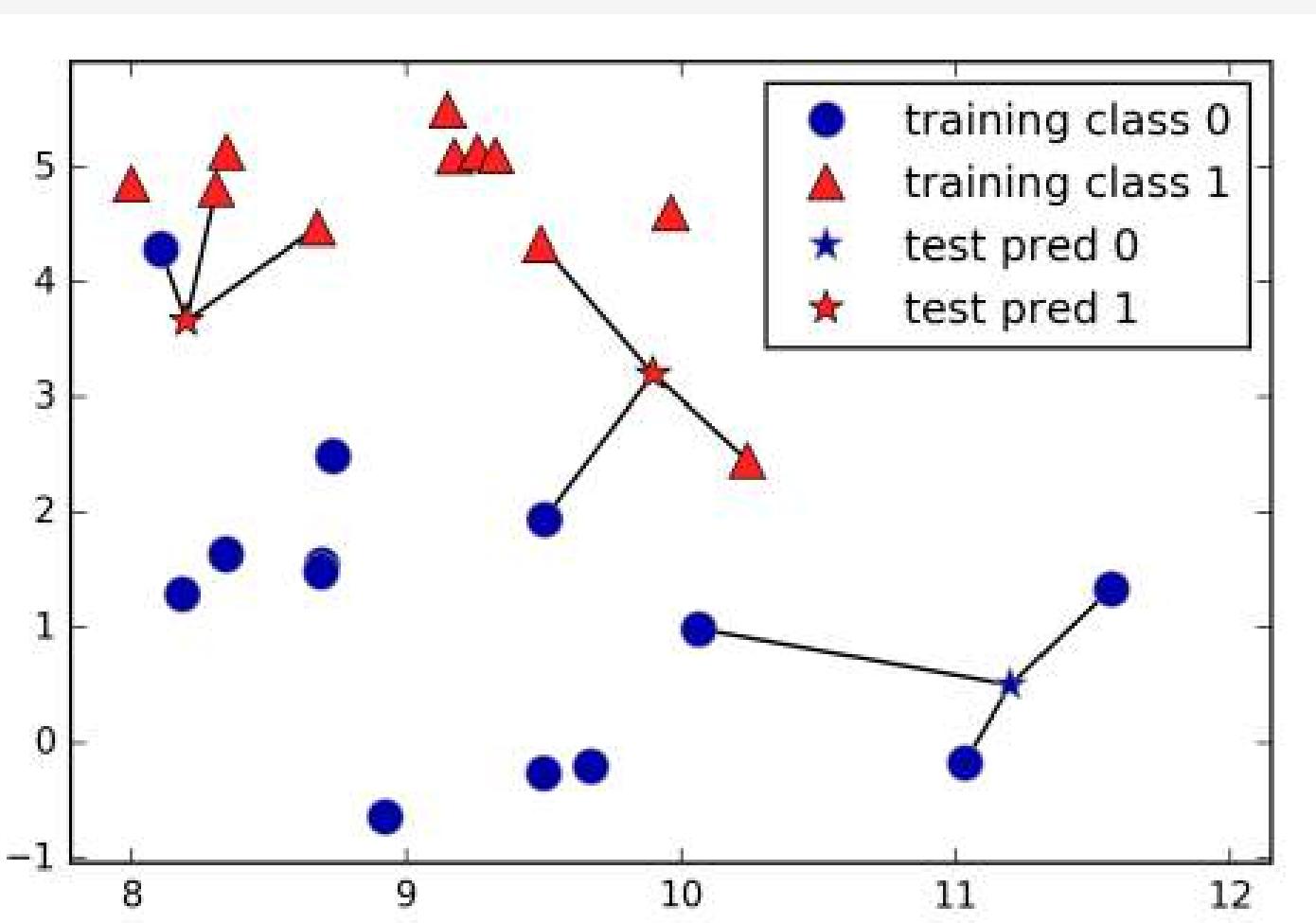
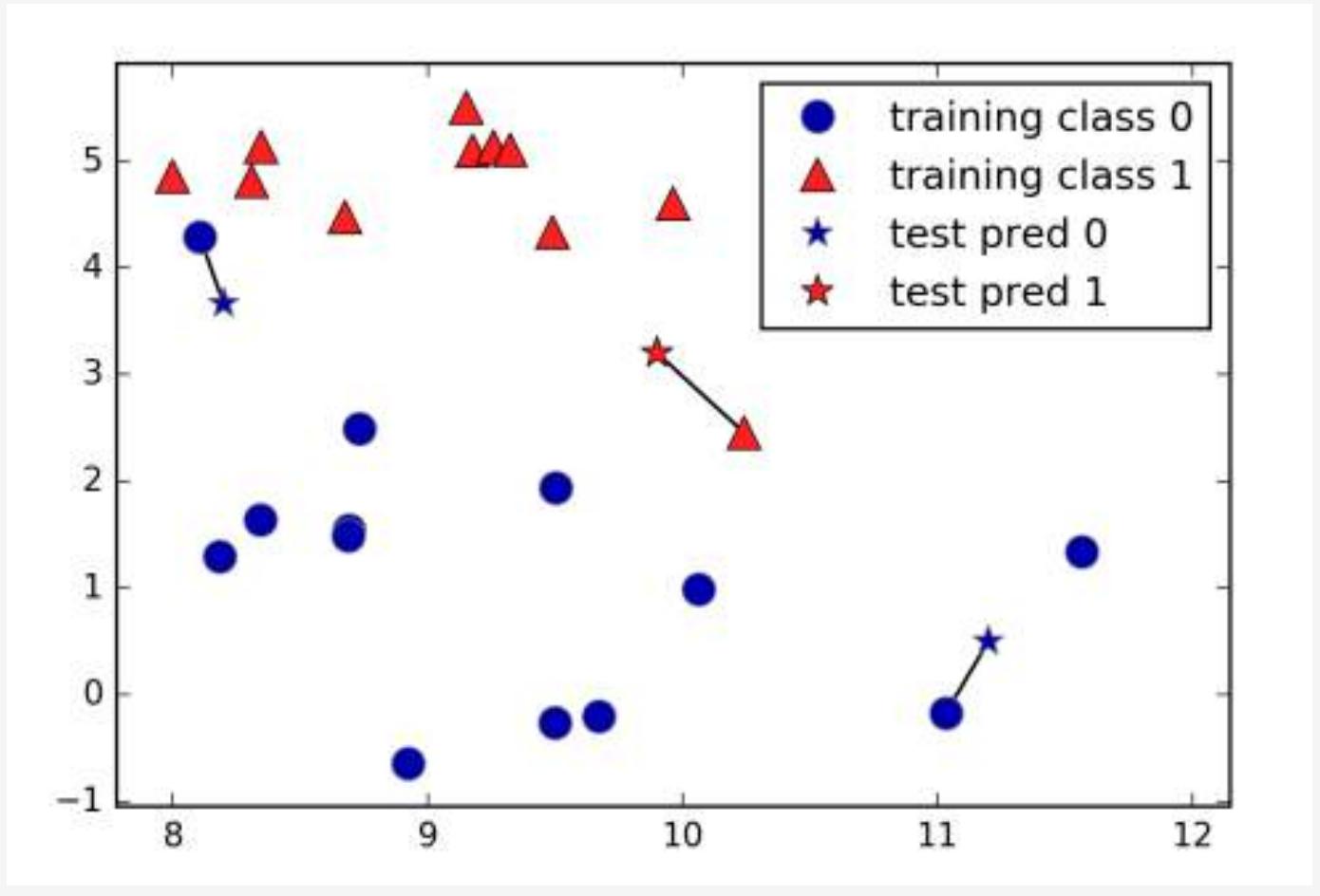
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

```
from sklearn.metrics import classification_report  
print(classification_report(y_test, pred_most_frequent,  
                             target_names=["not nine", "nine"]))
```

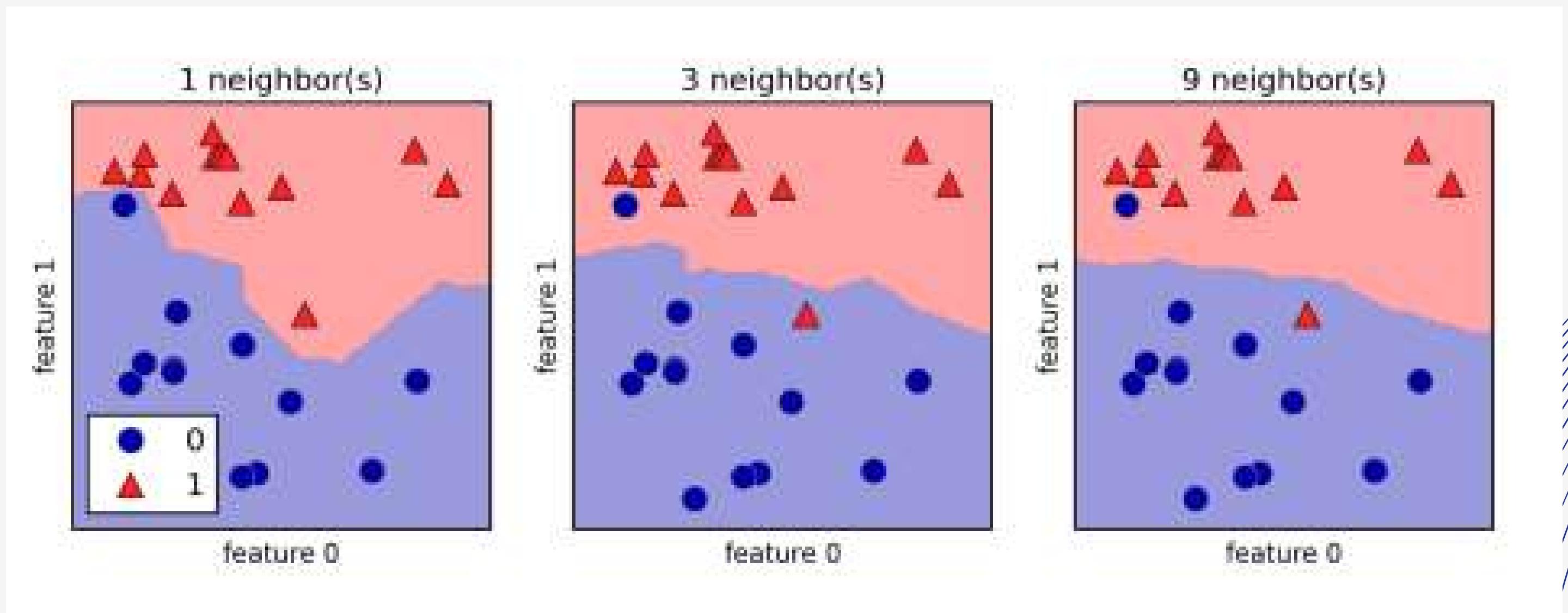
	precision	recall	f1-score	support
not nine	0.90	1.00	0.94	483
nine	0.00	0.00	0.00	47
avg / total	0.80	0.90	0.85	450

KNN





داده های گل آیریس



Decision Tree

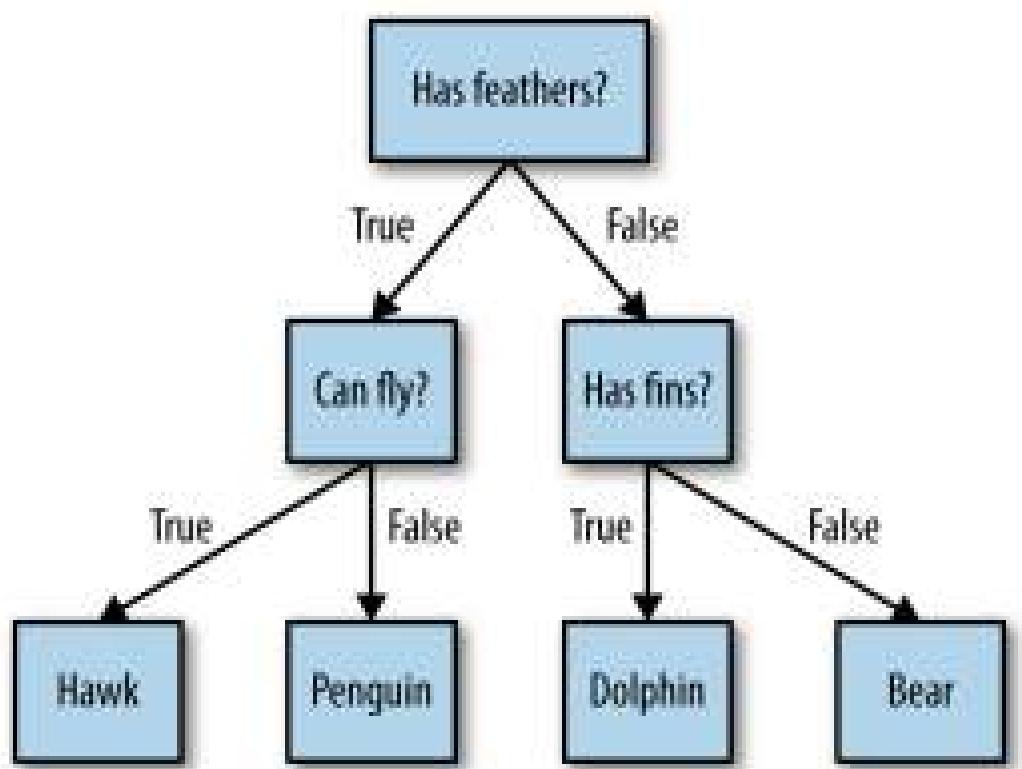


Figure 2-22. A decision tree to distinguish among several animals

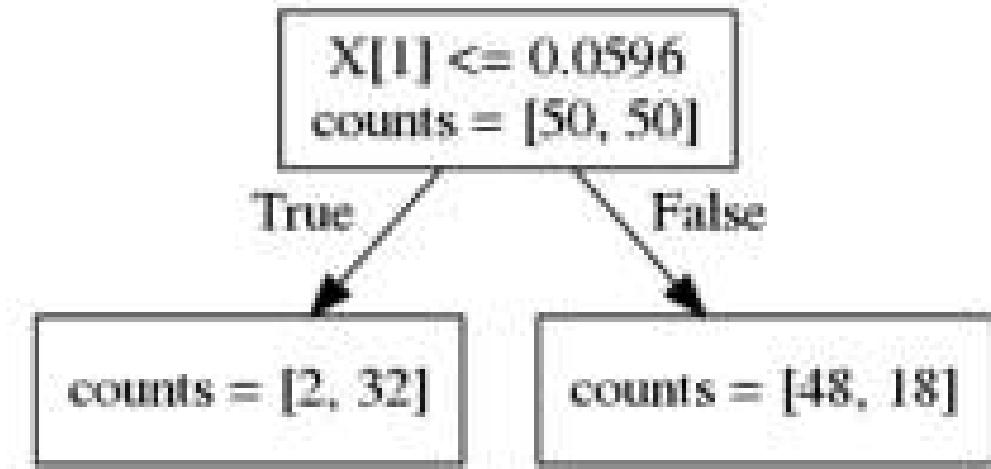
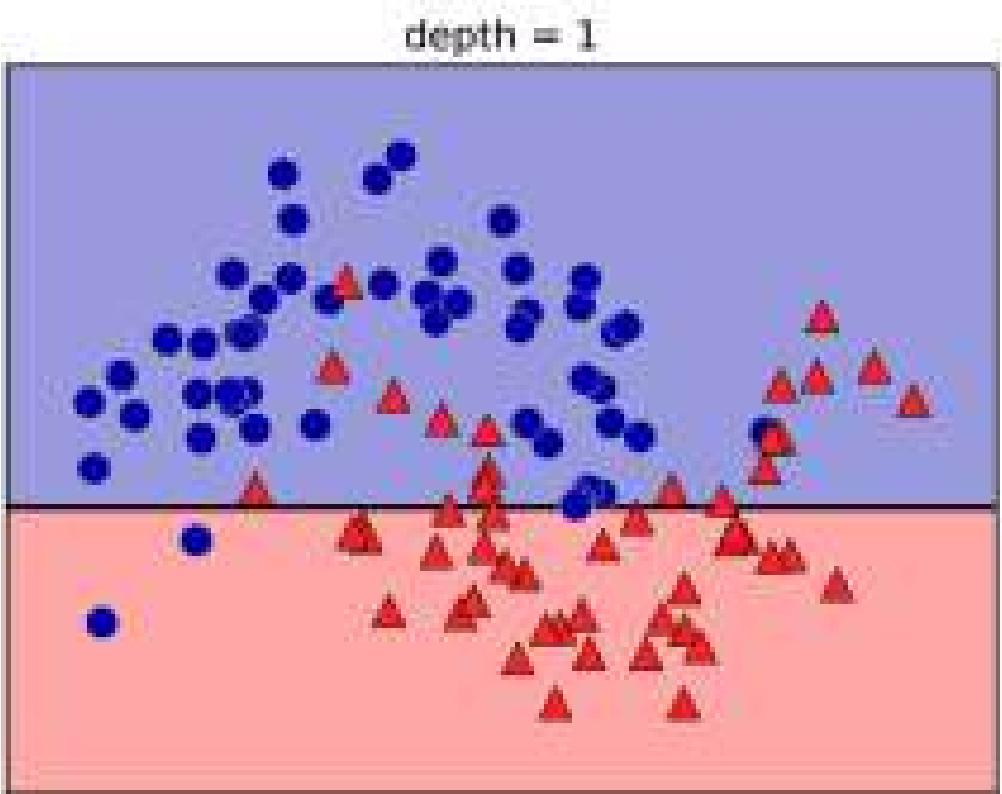
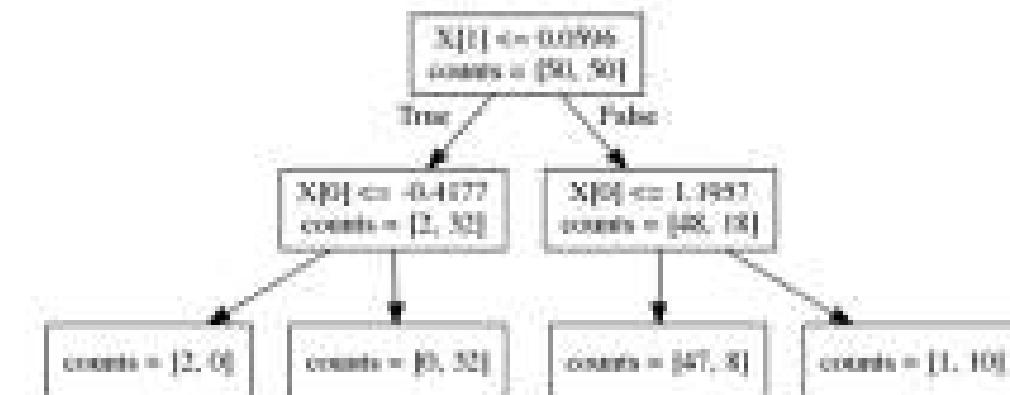
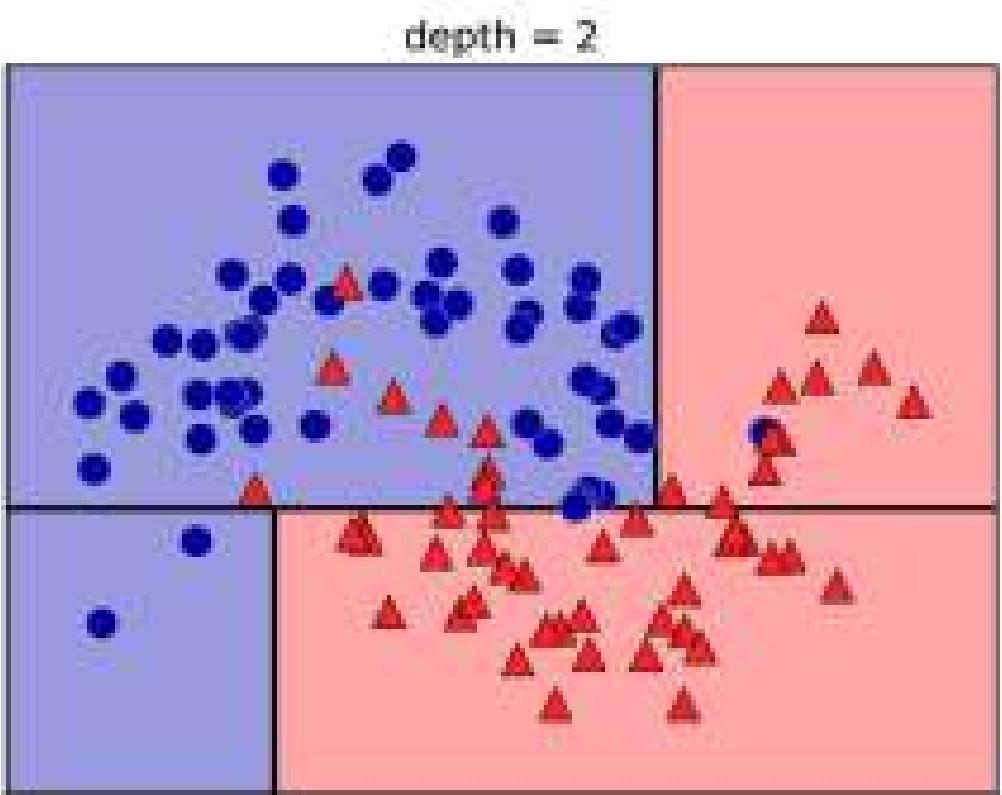
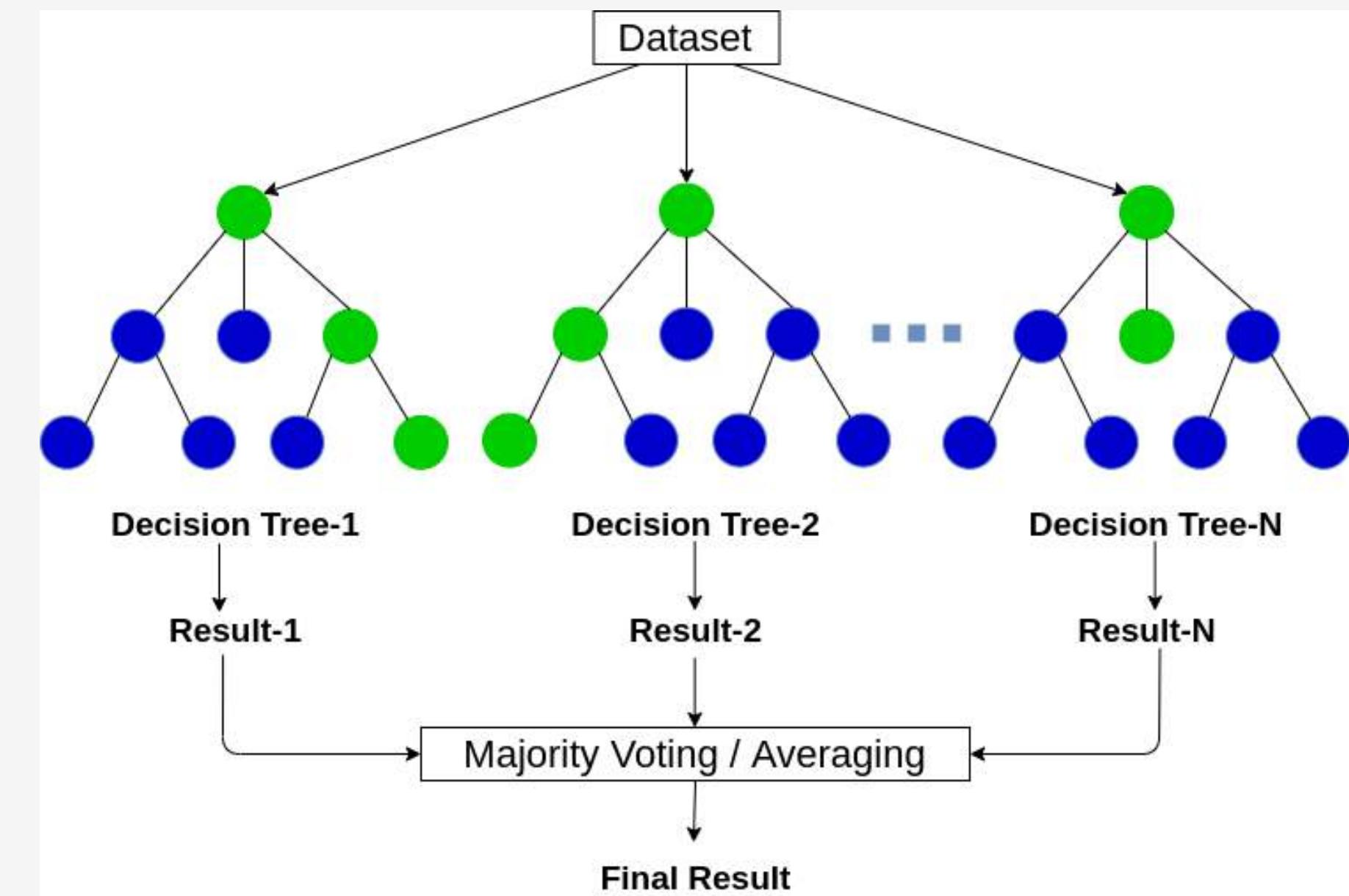


Figure 2-24. Decision boundary of tree with depth 1 (left) and corresponding tree (right)

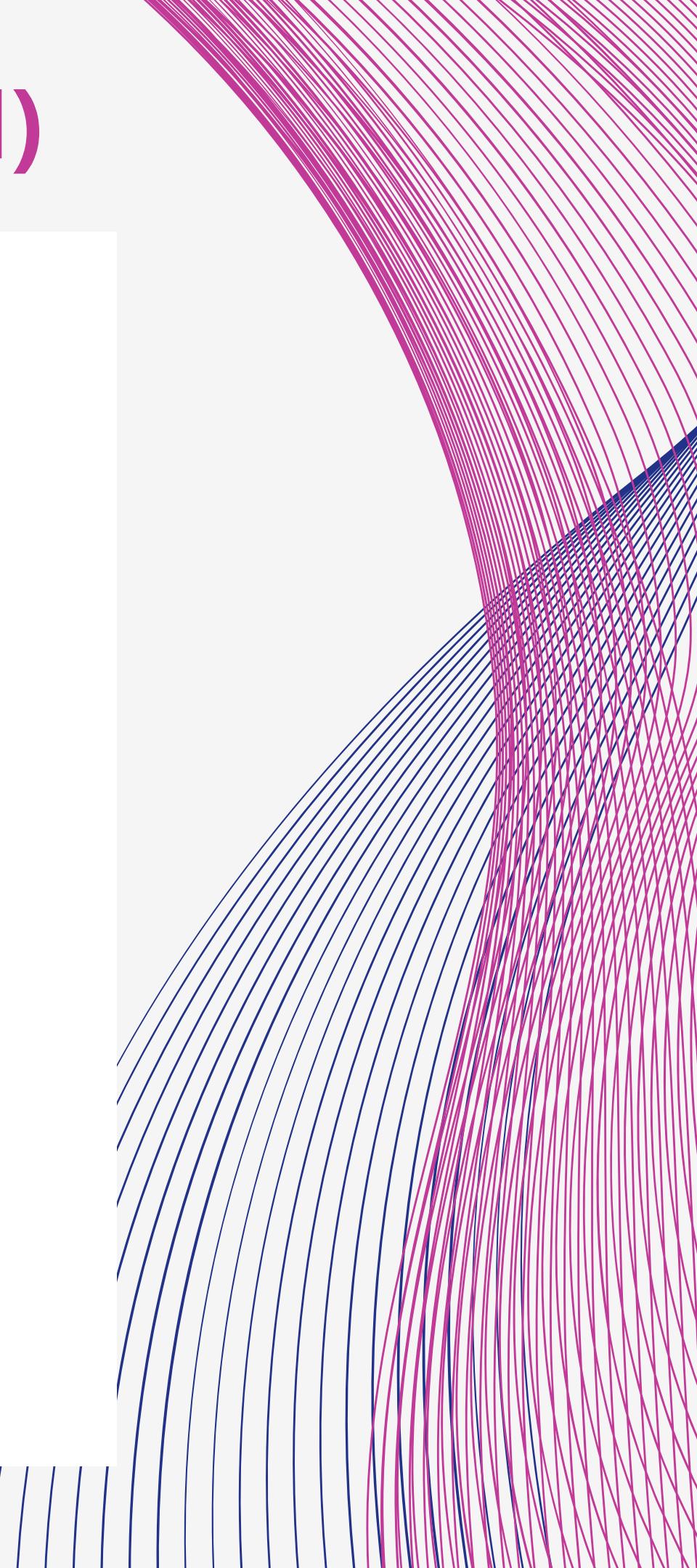
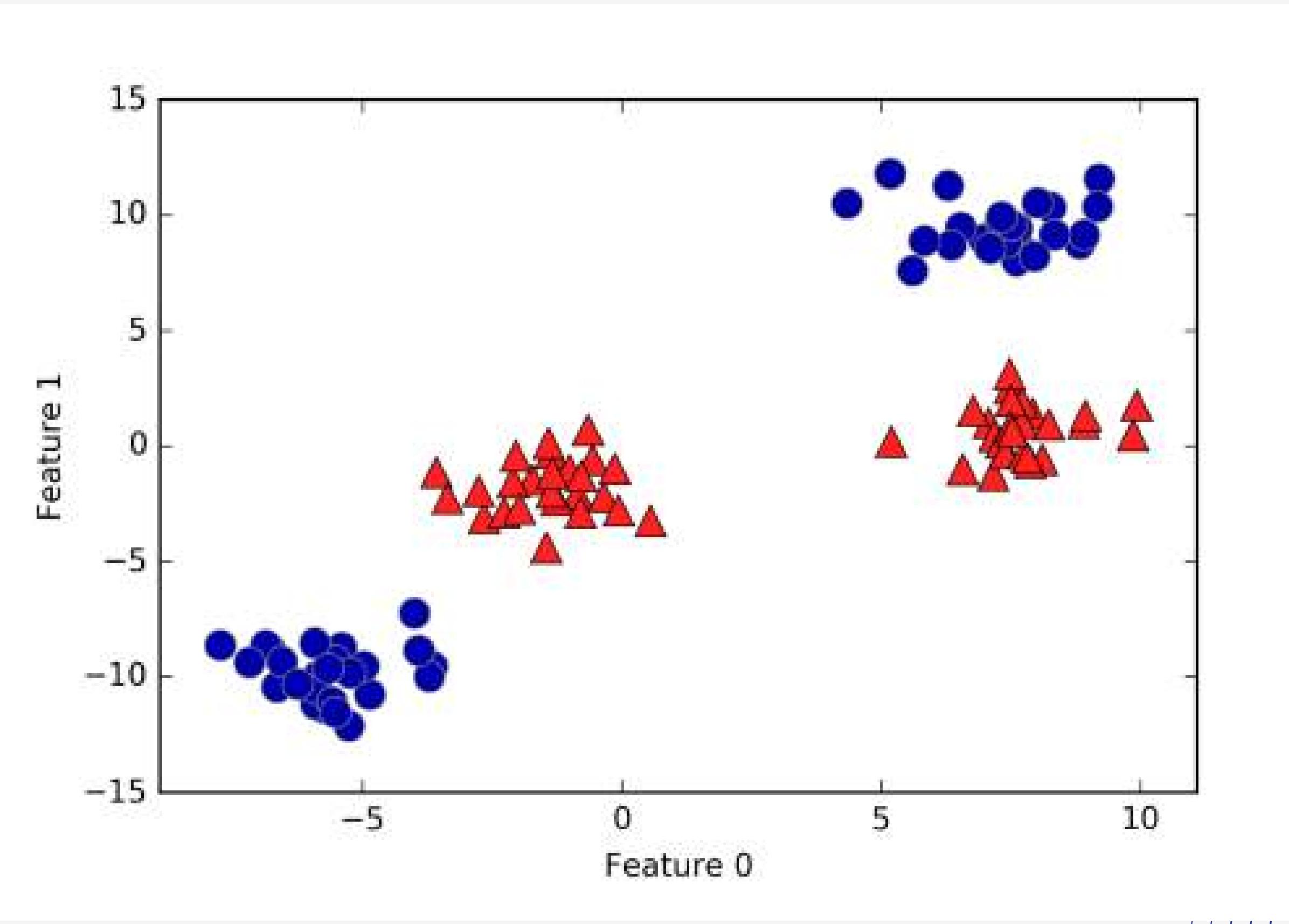


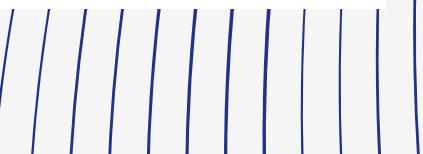
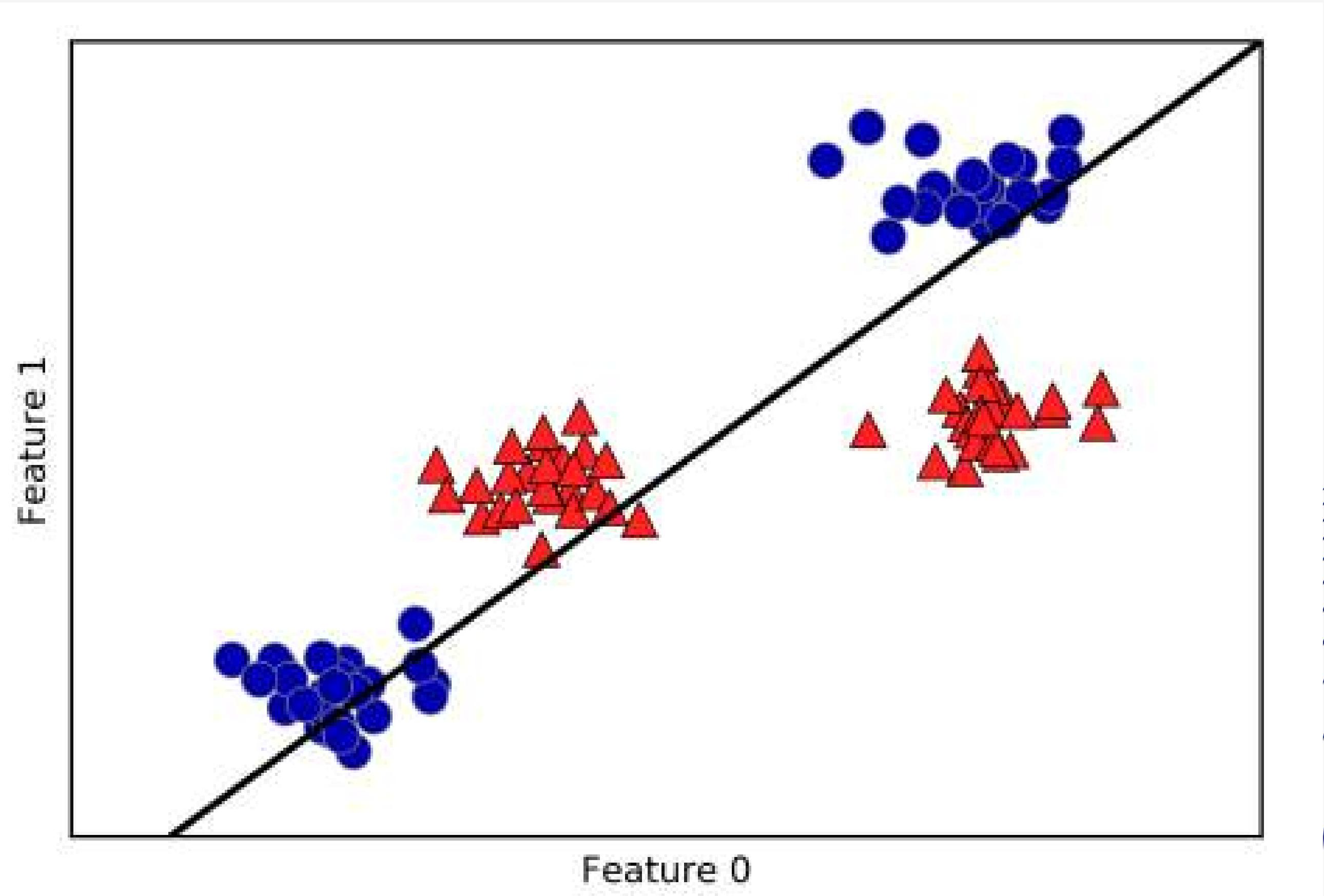


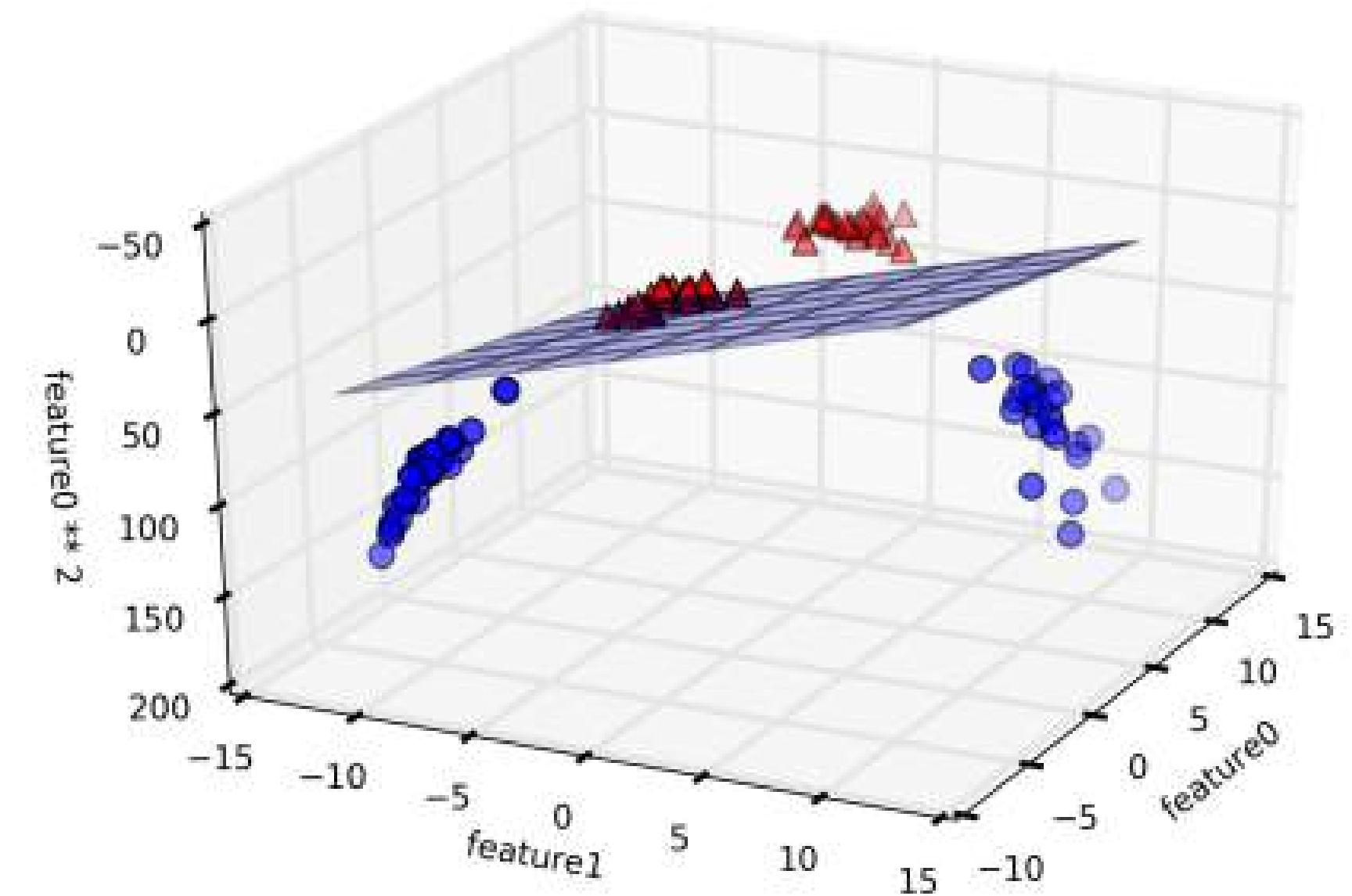
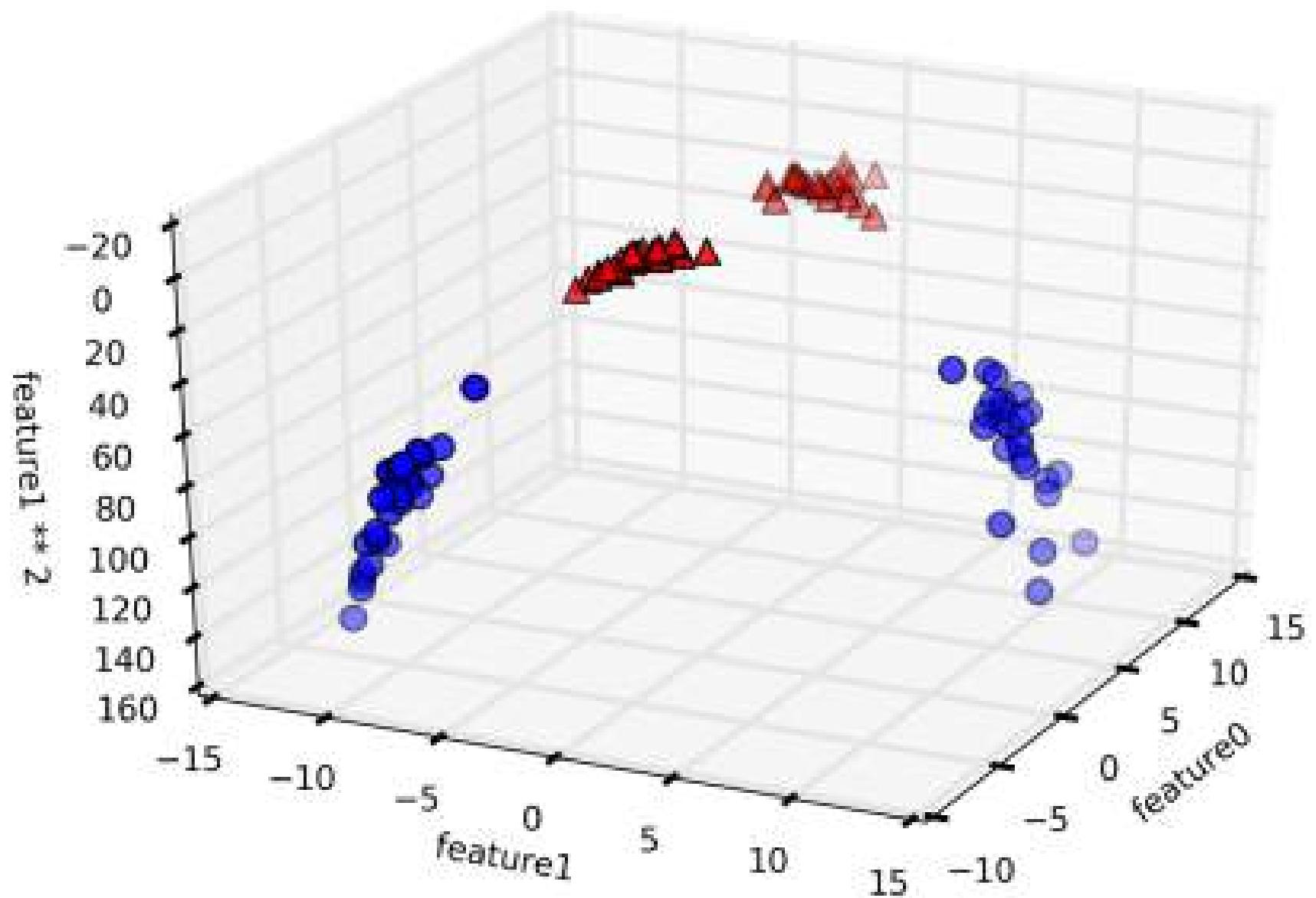
RANDOM FOREST



Support vector Machine (SVM)







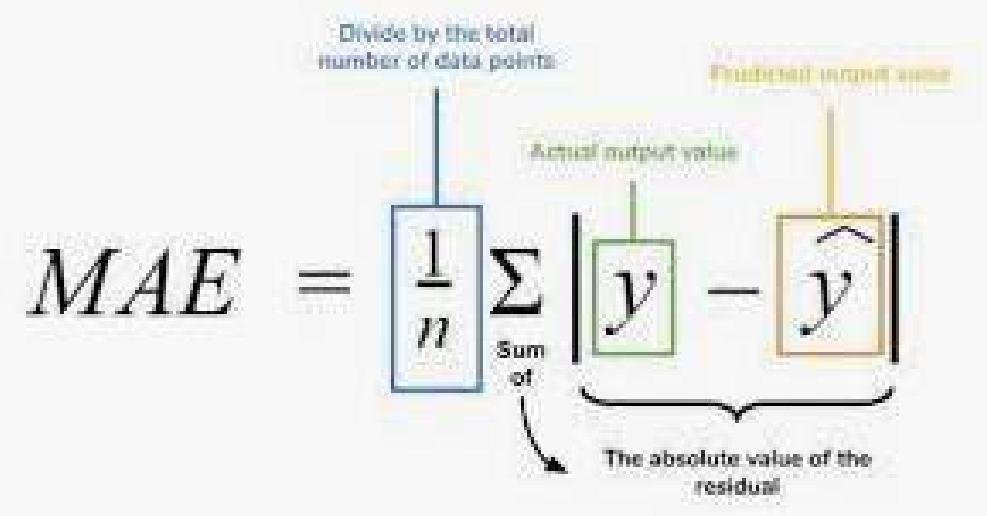
METRICS FOR REGRESSION

Regression

'explained_variance'	<code>metrics.explained_variance_score</code>
'max_error'	<code>metrics.max_error</code>
'neg_mean_absolute_error'	<code>metrics.mean_absolute_error</code>
'neg_mean_squared_error'	<code>metrics.mean_squared_error</code>
'neg_root_mean_squared_error'	<code>metrics.mean_squared_error</code>
'neg_mean_squared_log_error'	<code>metrics.mean_squared_log_error</code>
'neg_median_absolute_error'	<code>metrics.median_absolute_error</code>
'r2'	<code>metrics.r2_score</code>
'neg_mean_poisson_deviance'	<code>metrics.mean_poisson_deviance</code>
'neg_mean_gamma_deviance'	<code>metrics.mean_gamma_deviance</code>
'neg_mean_absolute_percentage_error'	<code>metrics.mean_absolute_percentage_error</code>
'd2_absolute_error_score'	<code>metrics.d2_absolute_error_score</code>
'd2_pinball_score'	<code>metrics.d2_pinball_score</code>
'd2_tweedie_score'	<code>metrics.d2_tweedie_score</code>

METRICS FOR REGRESSION

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \mu)^2}.$$

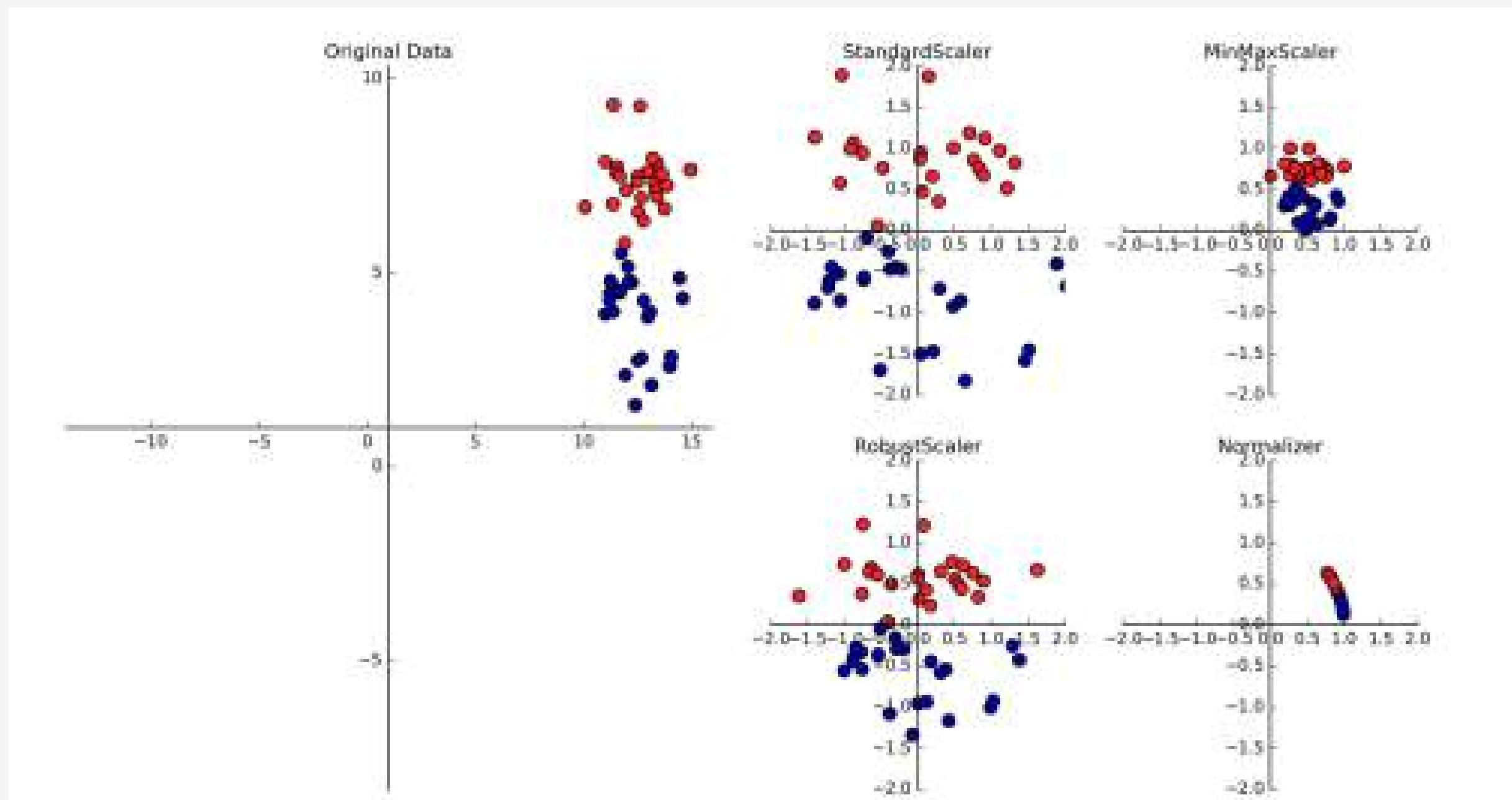


$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2$$

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - p_i)^2}$$

$$\text{Mean Absolute Percentage Error, MAPE} = \frac{\sum \left| \left(\frac{F_t - A_t}{A_t} \right) \cdot 100 \right|}{n}$$

Transformation and Scaler



Cross Validation (CV)

cross_val_score

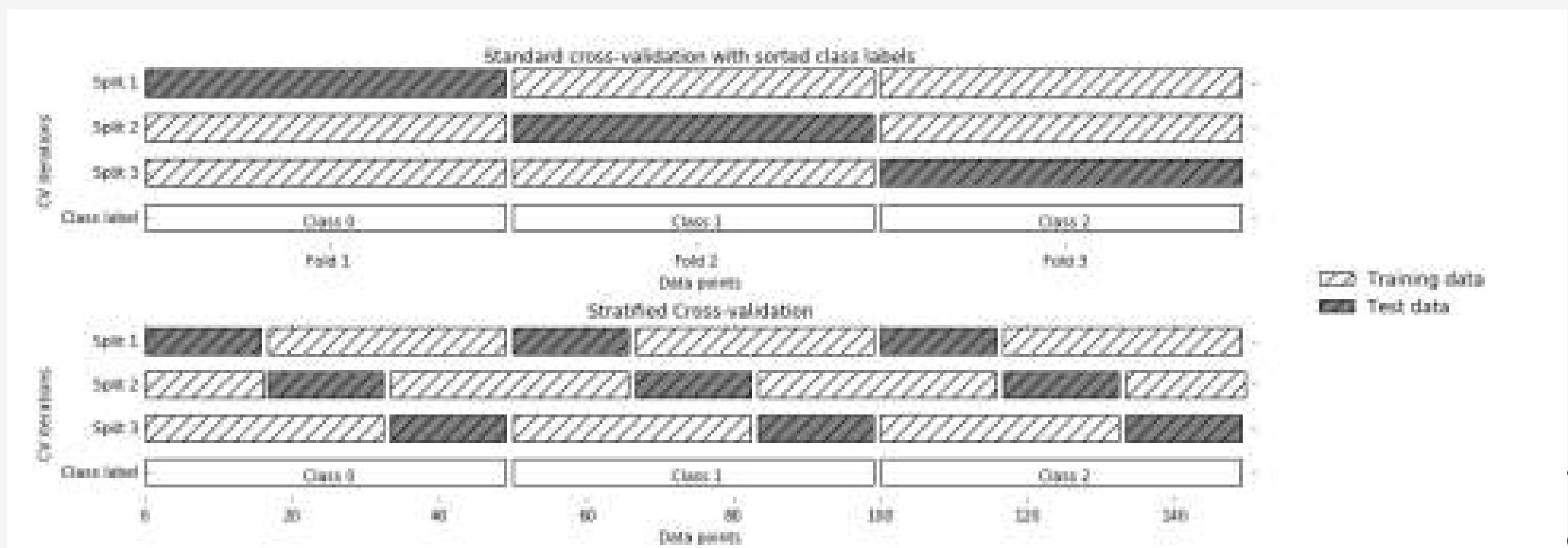
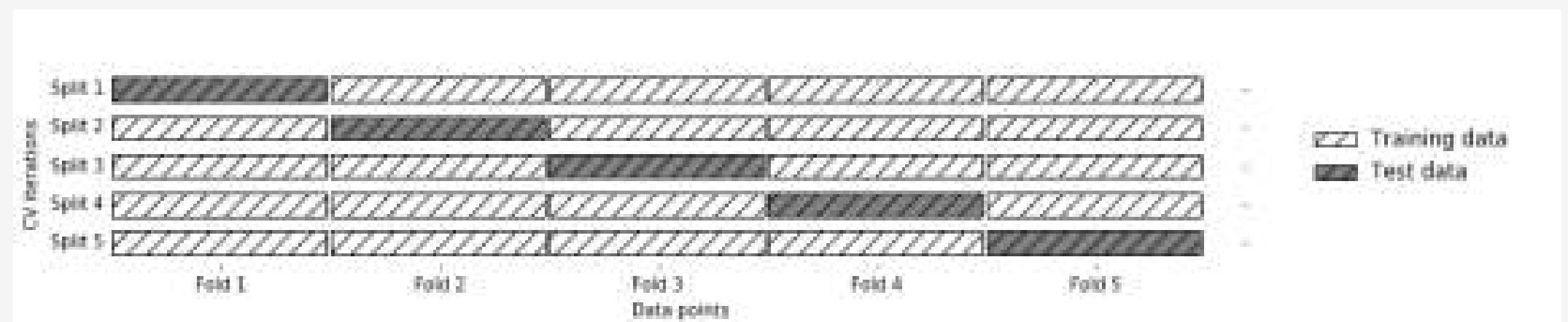
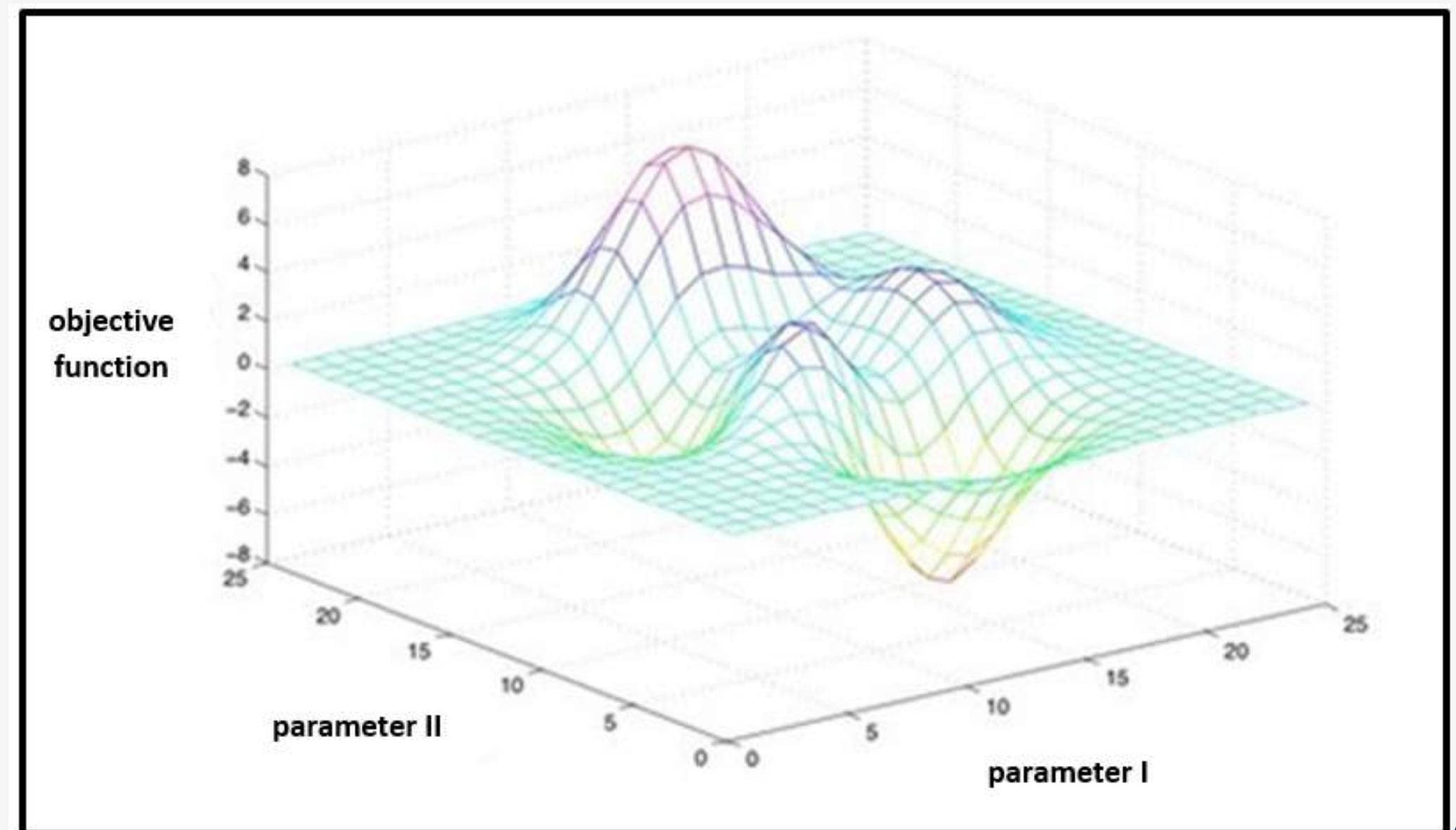
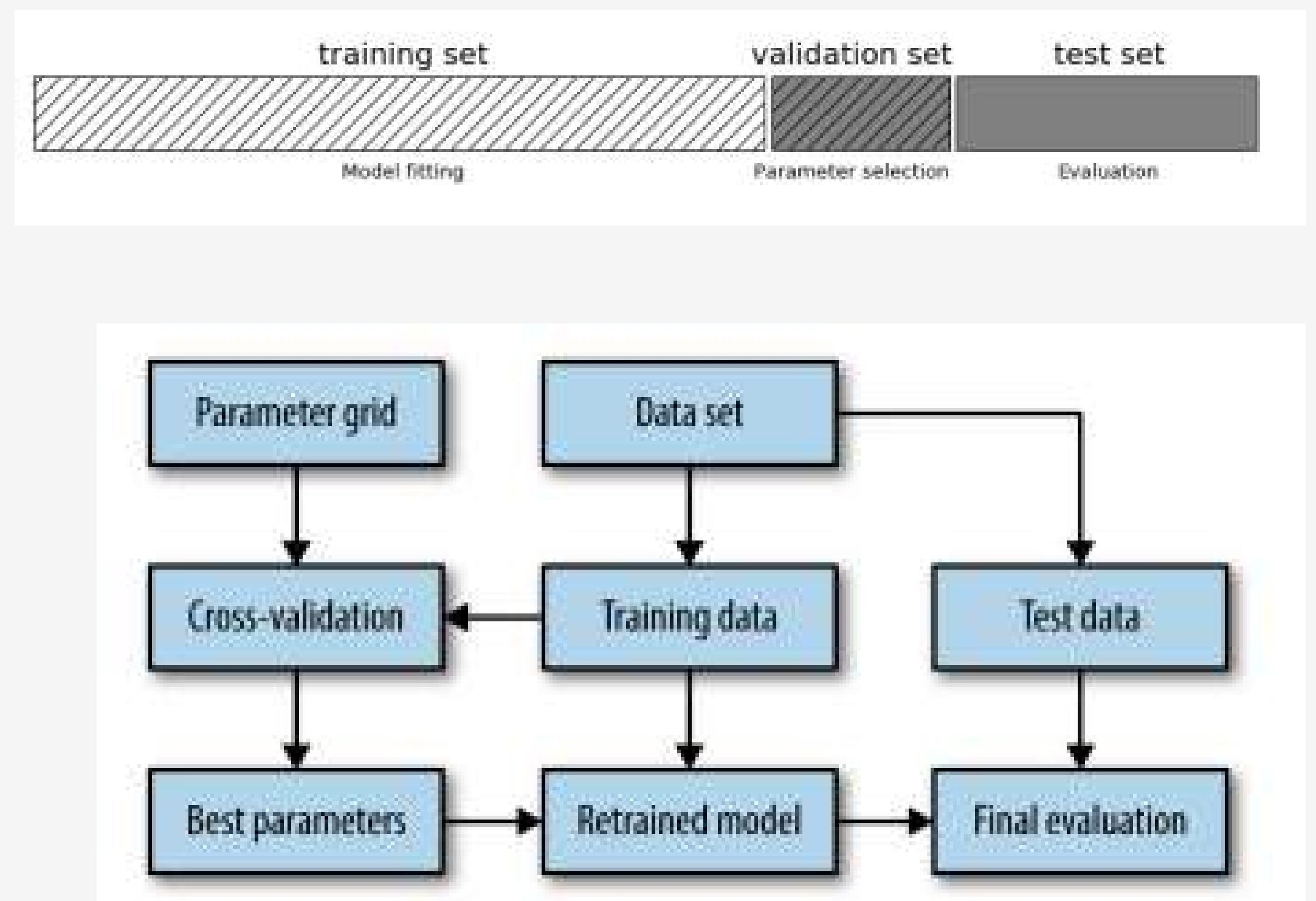


Figure 5-2. Comparison of standard cross-validation and stratified cross-validation when the data is ordered by class label

Grid Search



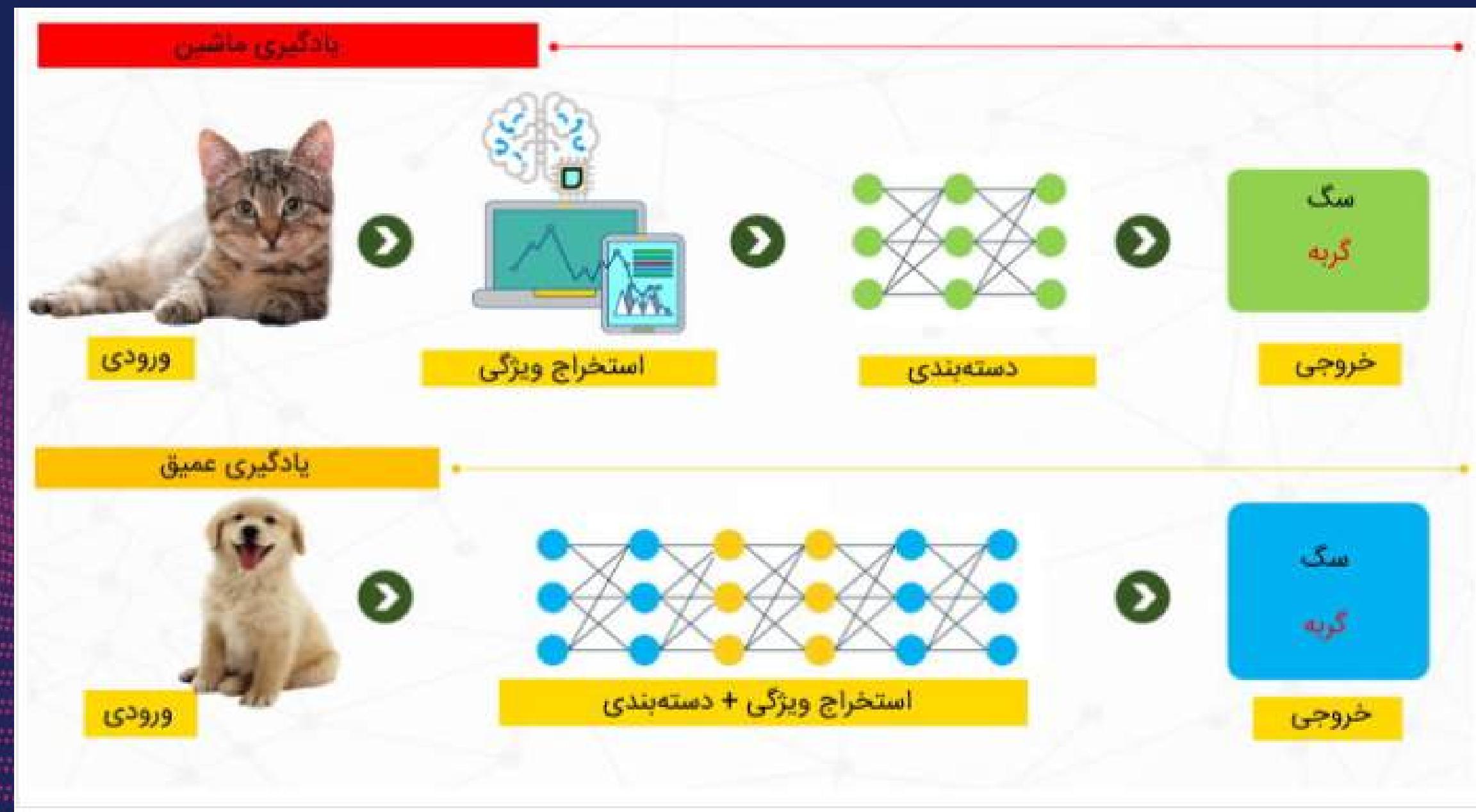
FINAL STRUCTURE



Section3: Intorduction to DEEP LEARNING

Ali Pilehvar Meibody
alipilehvar1999@gmail.com
ai.course22.alipilehvar@gmail.com

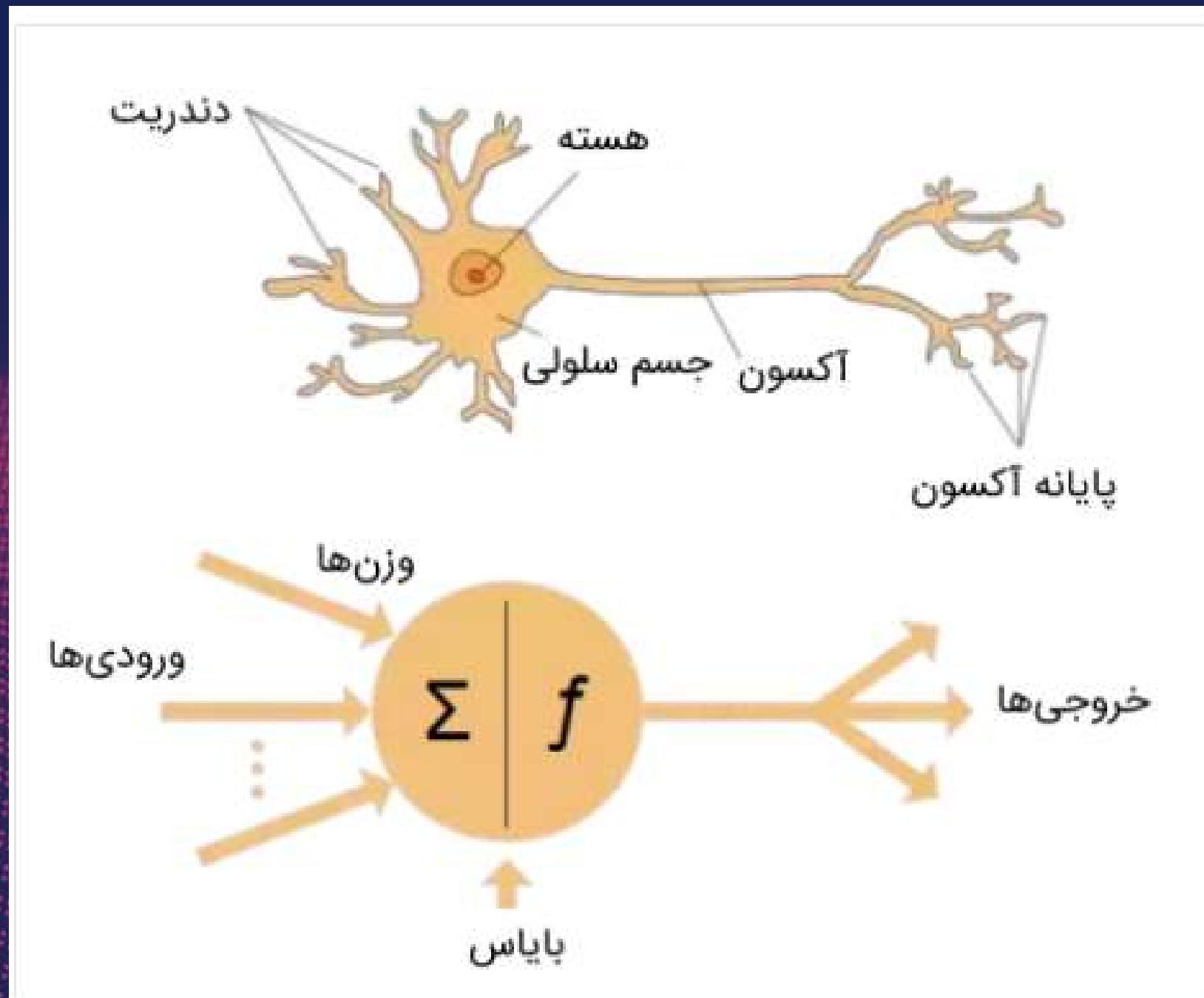
تفاوت یادگیری ماشین و یادگیری عمیق

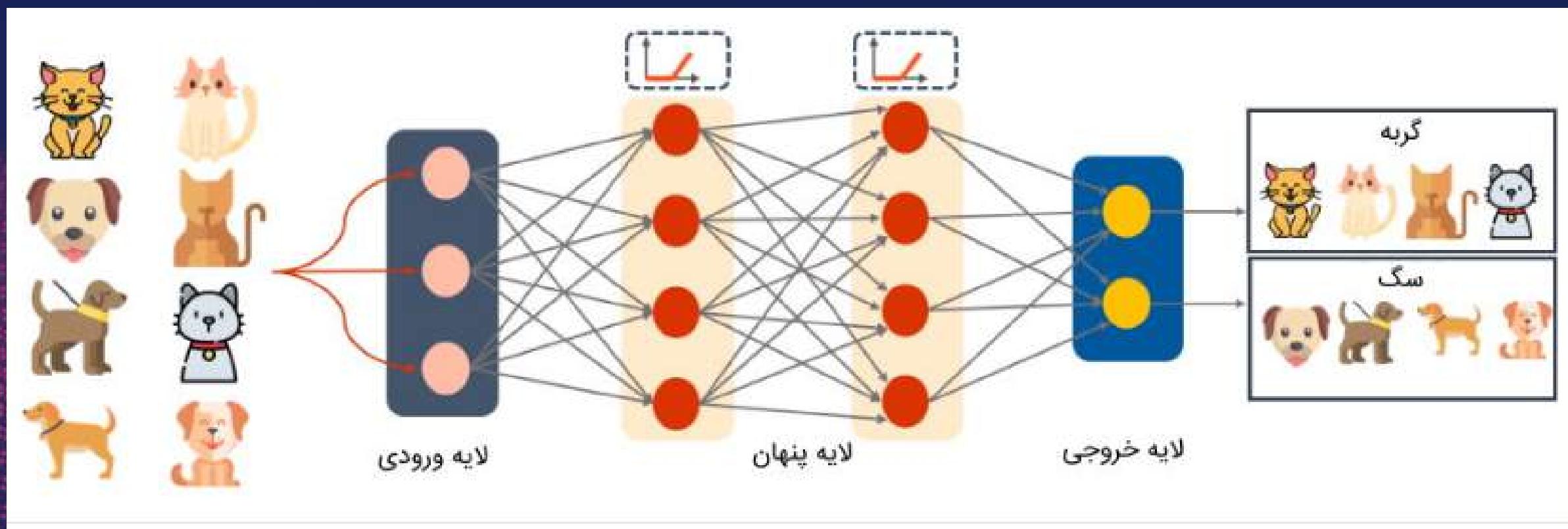


مدل ها

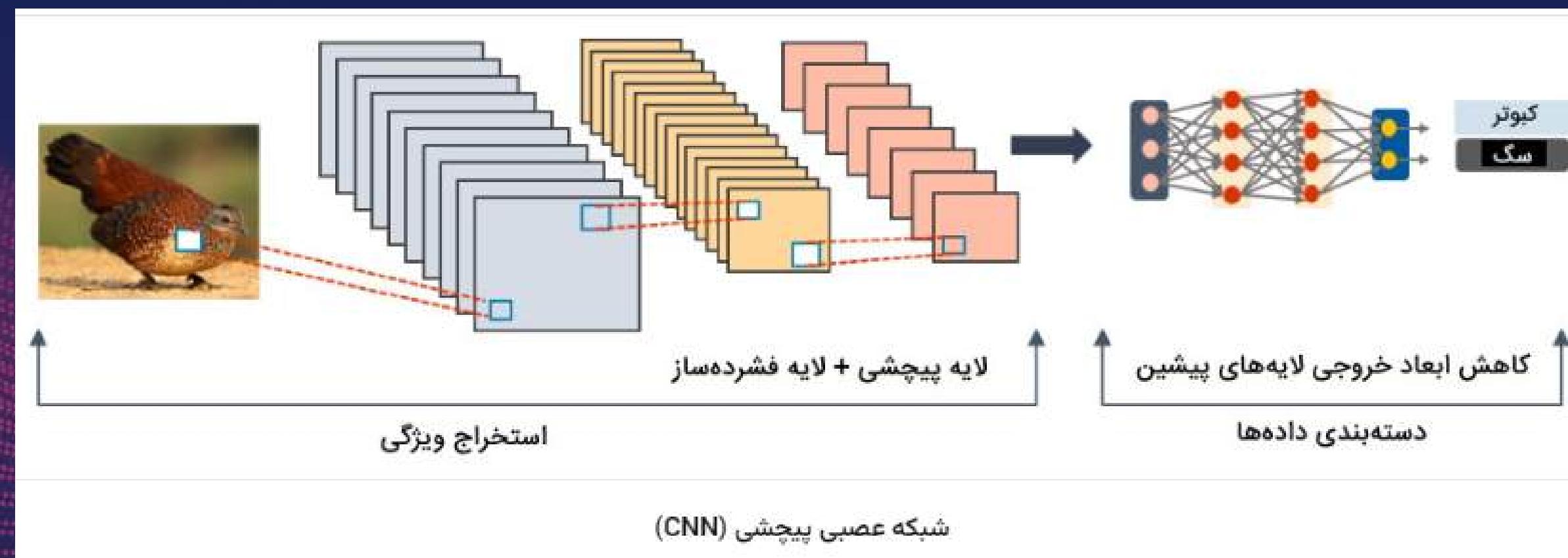
شبکه عصبی «چندلایه پرسپترون» (Multi Layer Perceptron | MLP)
«شبکه عصبی پیچشی» (Convolutional Neural Network | CNN)
«شبکه عصبی بازگشتی» (Recurrent Neural Network | RNN)
«شبکه عصبی حافظه طولانی کوتاه مدت» (Long Term-Short Memory | LSTM)
«شبکه مولد تخاصمی» (Generative Adversarial Network)
مدل «خودرمزنگذار» (Autoencoder)
مدل «ترسیم خودسازمان دهنده» (Self Organizing Map | SOM)

Multi layer Perceptron

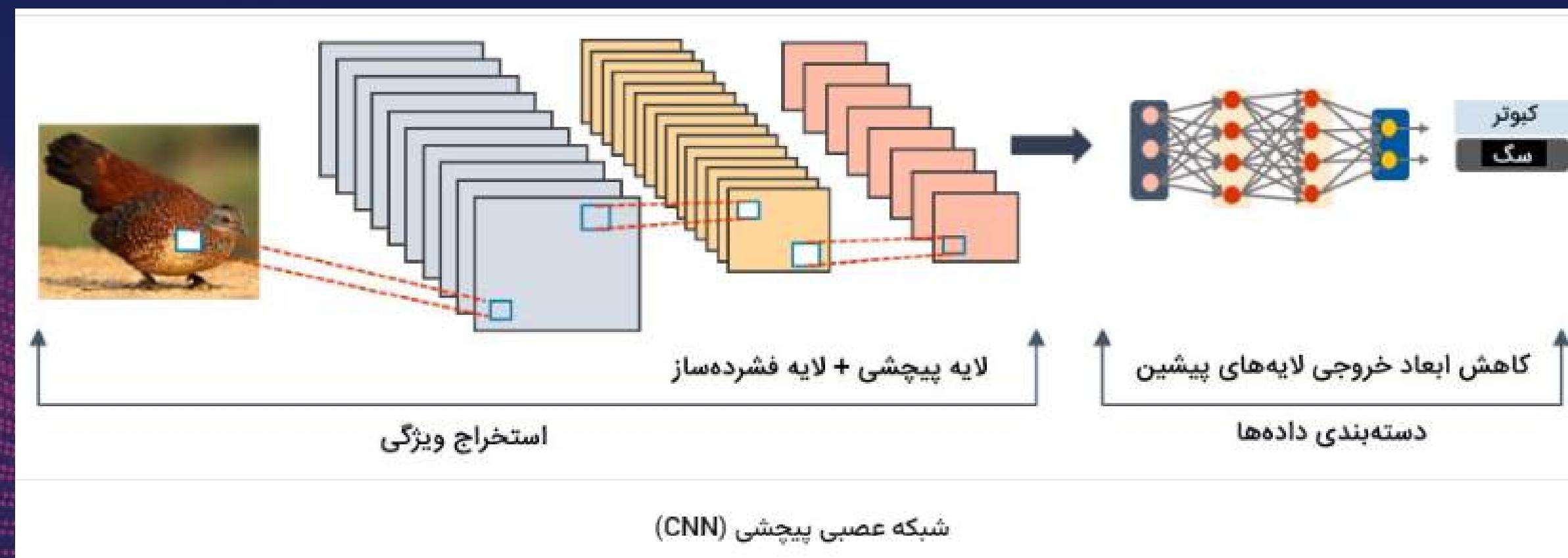




CNN



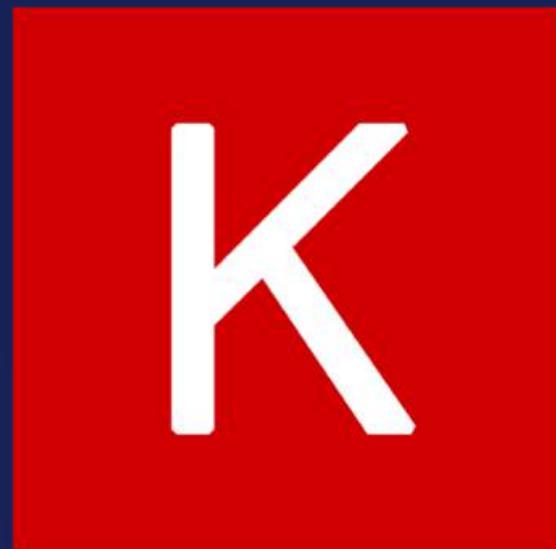
CNN



مفهومیت درین کتابخانه ها



TensorFlow



Keras

Section4: What We Do Next

Ali Pilehvar Meibody
alipilehvar1999@gmail.com
ai.course22.alipilehvar@gmail.com

- 1- دوره کردن تمام مطالب گذشته
- 2- نگاه به سایت w3school برای مطالب زبان پایتون
- 3- مراجعه به داکیومنتیشن های نامپای, پازداس, متپلاتلیپ
- 4- استفاده از داده های اسکای لرن برای تمرین بیشتر
- 5- خواندن کتاب introduction to machine learning
- 6- کار با داده های مقاله ها و مقایسه با ارور های آنها
- 7- شروع دیپ لرینگ با استفاده از کتاب و داکیومنتیشن سایت تنسورفلو

Gam Lab

تماس

Gamlab.aut.ac.ir



ai.course22.alipilehvar@gmail.com



09213529871

