



Maskininlärning

Installera Anaconda
Python

Nummergisslek

Vi ska bygga ett litet program, RIDER, som ska gissa vilket nummer som du tänker på. Reglerna är att du tänker på ett heltal mellan 1 och 100. RIDER gissar på ett tal och du säger om det är större, mindre eller rätt svar.

Skriv om till Pythonkod

```
min := 1
max := 100
finished := FALSE;
WHILE (!finished)
    myGuess := a random number between min and max
    PRINT 'Is it myGuess?'
    response := text from user input
    IF (response = 'correct')
        finished = TRUE

    ELSE IF (response = 'too large')
        max := myGuess - 1

    ELSE IF (response = 'too small')
        min := myGuess + 1

PRINT 'Yay!'
```

Superenkel bot

```
name = input("Hej! Vad heter du?")
age = int(input("Hej " + name + "! Vad trevligt att träffa
dig! Jag heter ACE. Hur gammal är du?"))
if age < 18:
    print("Vilken liten plutt du är!")
else:
    print("Vilken gamling du är!")
```

Uppgift: Utveckla boten exempelvis genom att lägga till en ny fråga som ger ett svar.

Python - repl.it

Python-programmering

1. Skriv ett program som funkar som en tärning dvs. skriver ut ett slumpat tal mellan 1 och 6.
2. Gör ett program där du kan mata in ett tal.
 - a. Lägg till en koll - Om talet är större än 5, skriv ut detta.
 - b. Lägg till ytterligare en koll - Om talet är lika med 5, skriv ut detta.
 - c. Om inget av ovanstående gäller, skriv ut att talet är mindre än 5.
 - d. Se till att ditt program upprepas i all oändlighet.

Python-programmering

3. Inför ett event har vi en lista sparad med åldrar på personerna som ska vara med: [14, 25, 43, 23, 55, 21, 13, 12, 17, 37, 18, 24, 15, 38, 23, 34, 13, 14].

- a. Gör ett program med en for-slinga som skriver ut allt som står i listan.
- b. Gör om programmet så att det räknar och skriver ut hur många som ska vara med.
- c. Gör om programmet så att det bara räknar och skriver ut hur många som är under 18.

Python-programmering

- Print-satser
- Matematiska operationer
- Variabler
- Inmatning
- If-satser (if, else och else if)
- Listor (Med tal och strängar)

- Random

```
import random  
print(random.randint(0,9  
))
```

Skriver ut slumpmässiga heltal mellan 0 och 9.
- for-slinga
- while-slinga

Anaconda Prompt

- `h:`
- `dir`
- `cd`
- `cd ..`
- `python programkod.py`

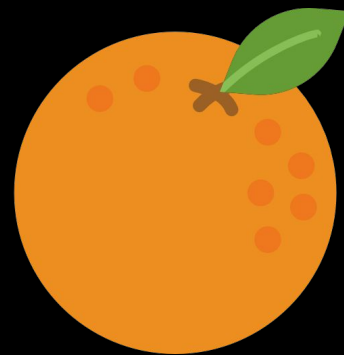
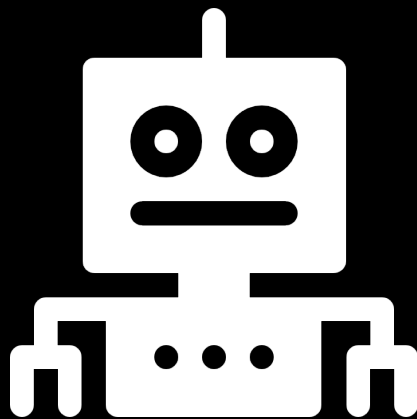
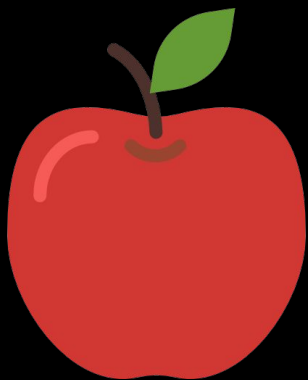
Inställningar

```
conda install python=3.6
```

```
pip install sklearn
```

Code writer





Vikt

Yta (slät, skrovlig)

AI



Maskin-
inlärning

Förbestämda
regler



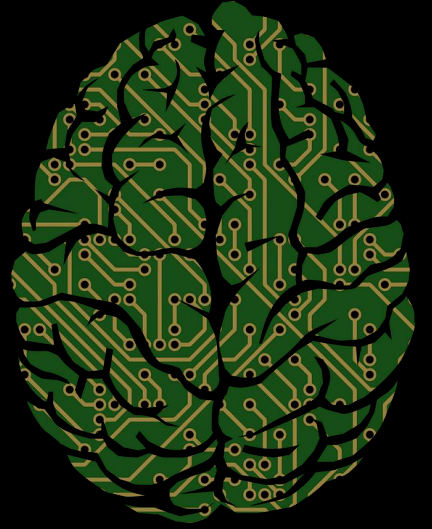
Djup-
inlärning

Beslutsträd

ELIZA

Maskininlärning

1. Samla in data (information)
2. Anpassa data
3. Välj modell (djupinlärning, beslutsträd, mm.)
4. Träning
5. Evaluera
6. Finjustera
7. Testa mot riktigt data

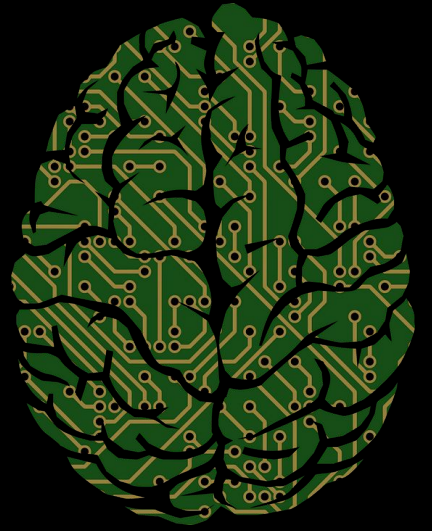


Djupinlärning

Neurala nätverk

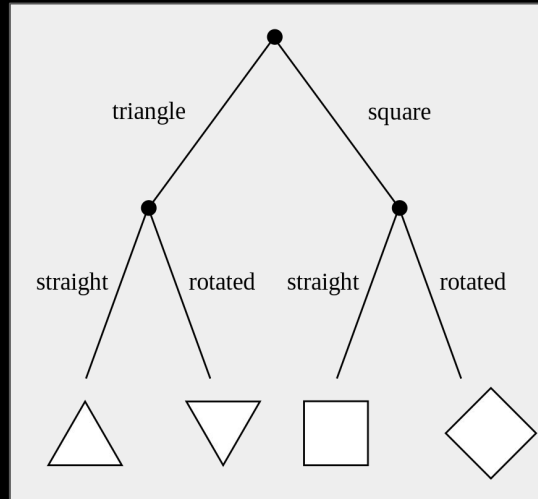
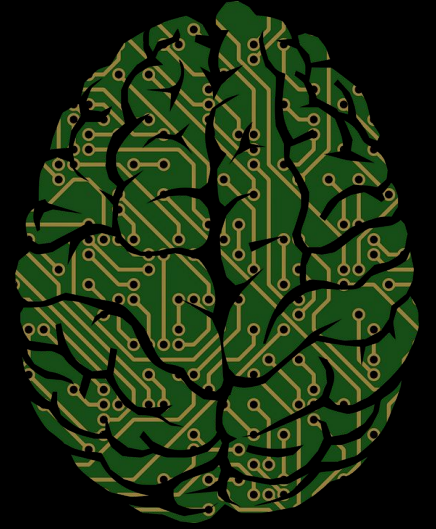
Påminner om hur hjärnan fungerar

<https://playground.tensorflow.org/>

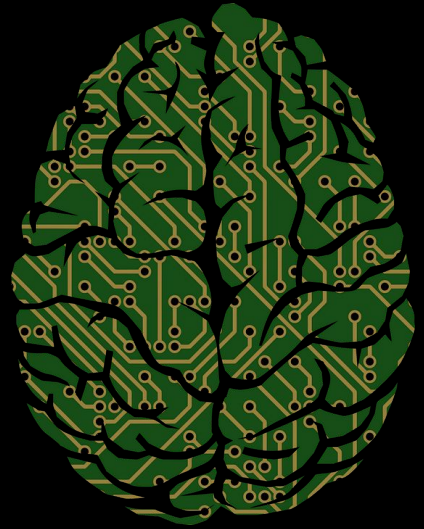


Beslutsträd

Beskriver olika alternativ beroende vilken information som ges.



1. Samla in data



| Frukt | Vikt | Yta |
|--------------|-------------|------------|
| Äpple | 140 | Slät |
| Äpple | 130 | Slät |
| Äpple | 145 | Slät |
| Apelsin | 150 | Skrovlig |
| Apelsin | 165 | Skrovlig |
| Apelsin | 170 | Skrovlig |

2. Anpassa data

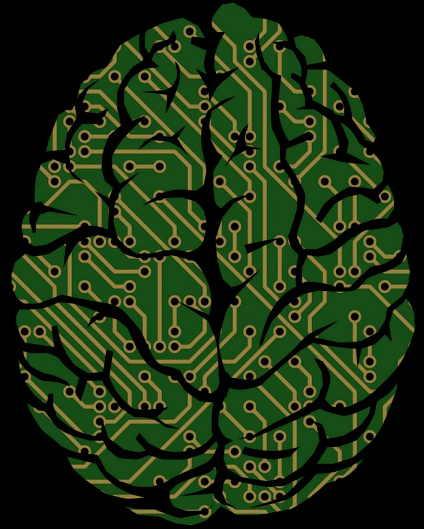
Skriv om datat så att datorn förstår

Äpple → 0

Apelsin → 1

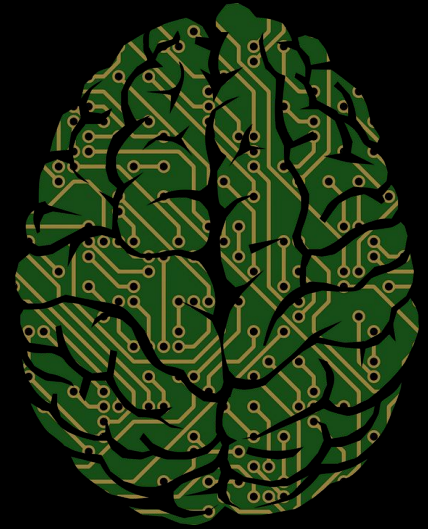
Skrovlig → 0

Slät → 1



2. Anpassa data

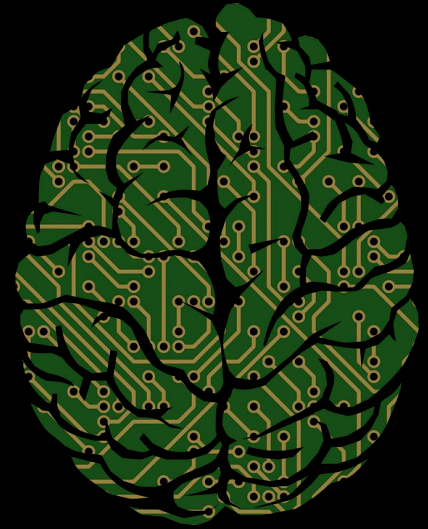
| Frukt | Vikt | Yta |
|-------|------|-----|
| 0 | 140 | 1 |
| 0 | 130 | 1 |
| 0 | 145 | 1 |
| 1 | 150 | 0 |
| 1 | 135 | 0 |
| 1 | 170 | 0 |



2. Anpassa data

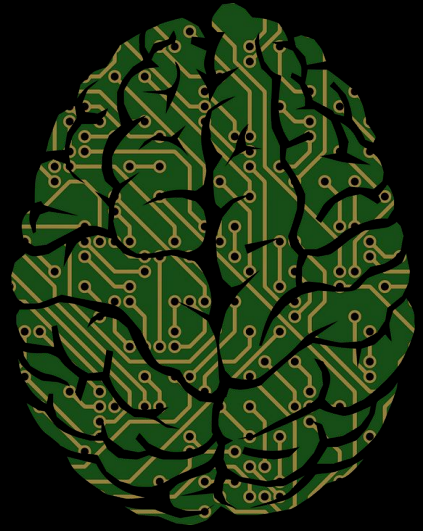
Välj ut testdata för evalueringen, steg 5

| Frukt | Vikt | Yta |
|-------|------|-----|
| 0 | 145 | 1 |
| 1 | 135 | 0 |



2. Anpassa data

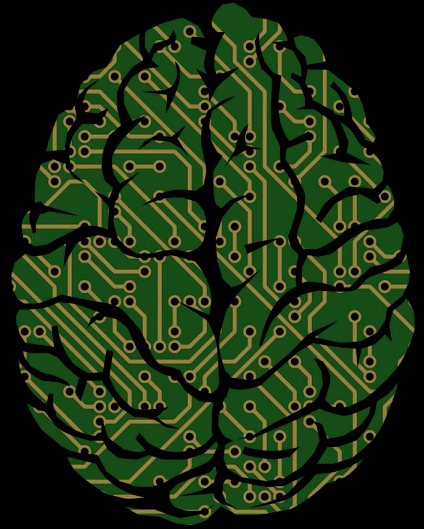
Färdig med träningsdata



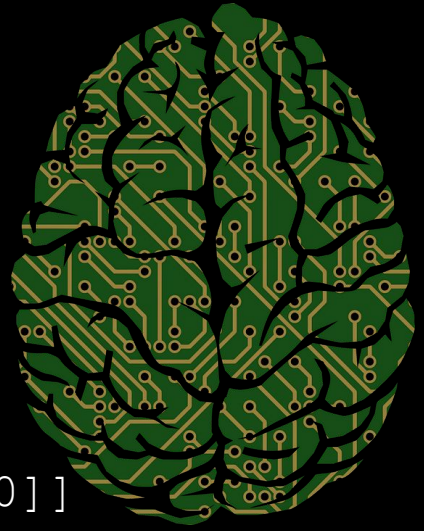
| Frukt | Vikt | Yta |
|-------|------|-----|
| 0 | 140 | 1 |
| 0 | 130 | 1 |
| 1 | 150 | 0 |
| 1 | 170 | 0 |

3. Välj modell

Python-bibliotek: sklearn
Beslutsträd



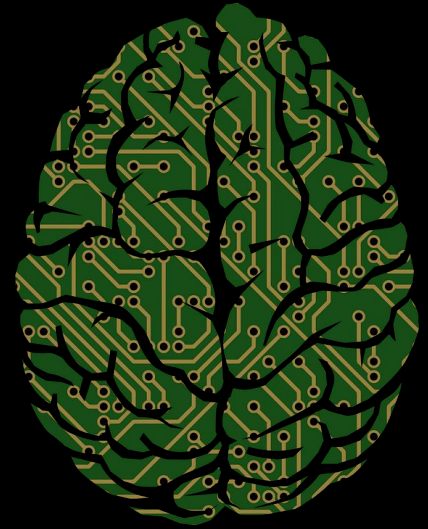
4. Träning



```
from sklearn import tree
features = [[140, 1], [130, 1], [150, 0], [170, 0]]
labels = [0, 0, 1, 1]
clf = tree.DecisionTreeClassifier()
clf = clf.fit(features, labels)
```


5. Evaluering

| Frukt | Vikt | Yta |
|-------------|------|--------------|
| 0 (äpple) | 145 | 1 (slät) |
| 1 (apelsin) | 165 | 0 (skrovlig) |

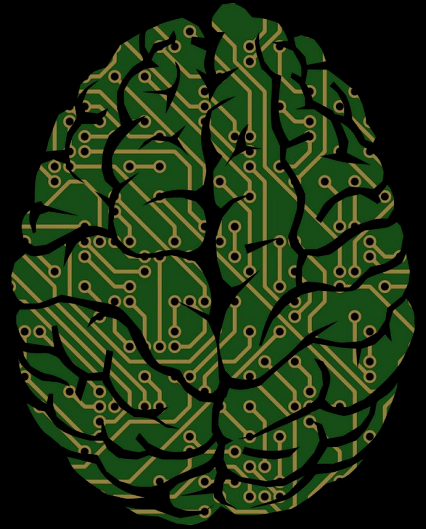


```
from sklearn import tree
features = [[140, 1], [130, 1], [150, 0], [170, 0]]
labels = [0, 0, 1, 1]
clf = tree.DecisionTreeClassifier()
clf = clf.fit(features, labels)
print(clf.predict([[145, 1]]) # [0] betyder äpple
print(clf.predict([[135, 0]]) # [1] betyder appelsin
```

6. Finjustering

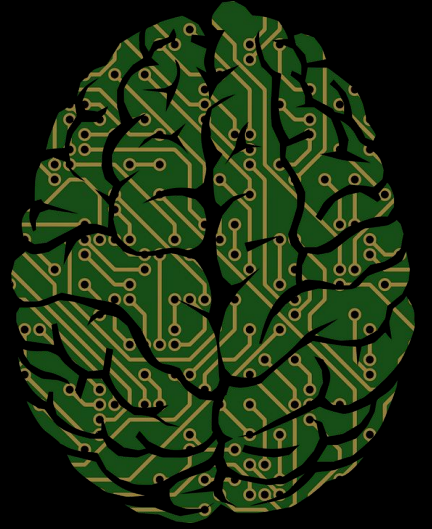
Hur kan vi förbättra programmet?

- Fler testfall
- Fler kategorier (tex färg, form)
- Modell



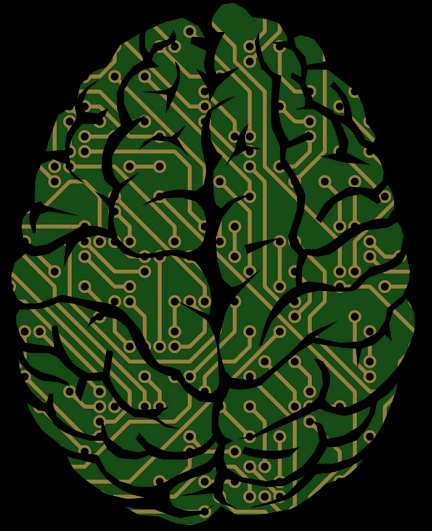
Maskininlärning

1. Samla in data (information)
2. Anpassa data
3. Välj modell (djupinlärning, beslutsträd, mm.)
4. Träning
5. Evaluera
6. Finjustera
7. Testa mot riktigt data



Uppgift

1. Ändra programmet så att utskriften blir tydligare. Programmet ska meddela om man har evaluerat en apelsin eller ett äpple istället för att svara med en siffra.



Ikoner hämtade från

flaticon.com (Smashicons, Freepik)