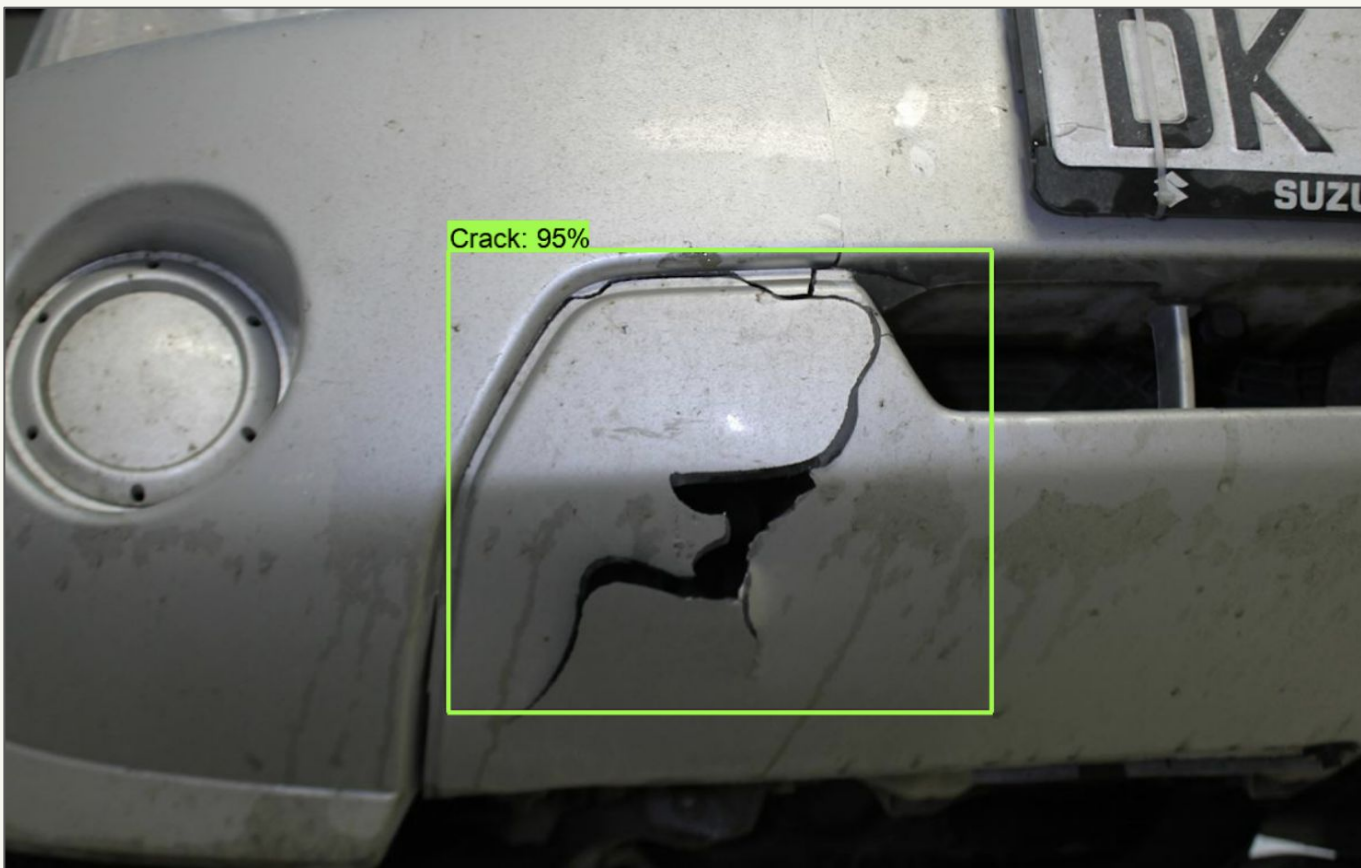
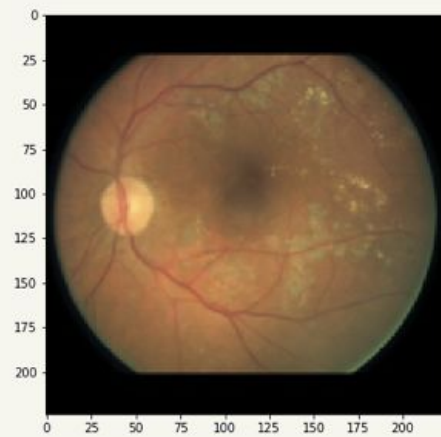
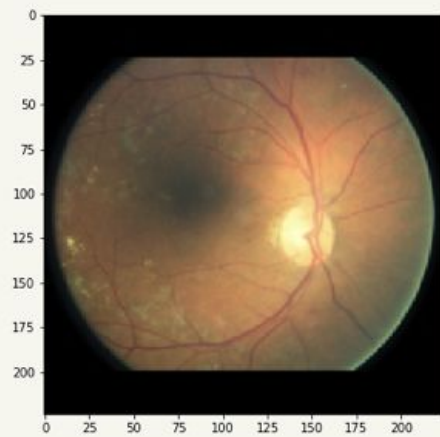
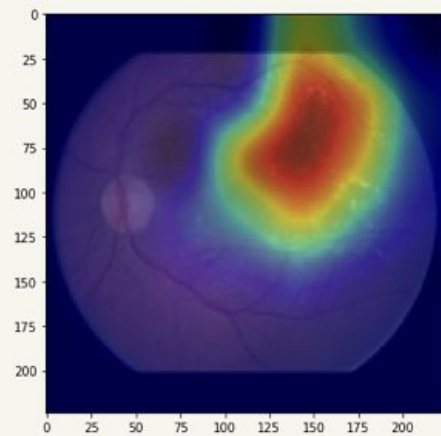
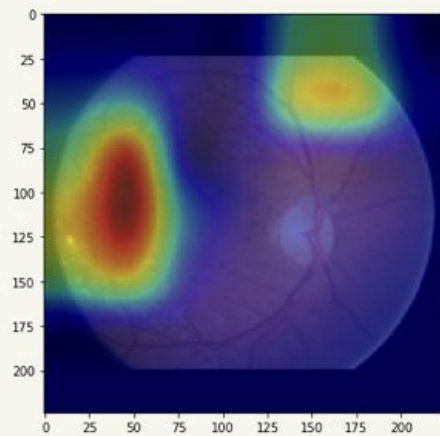


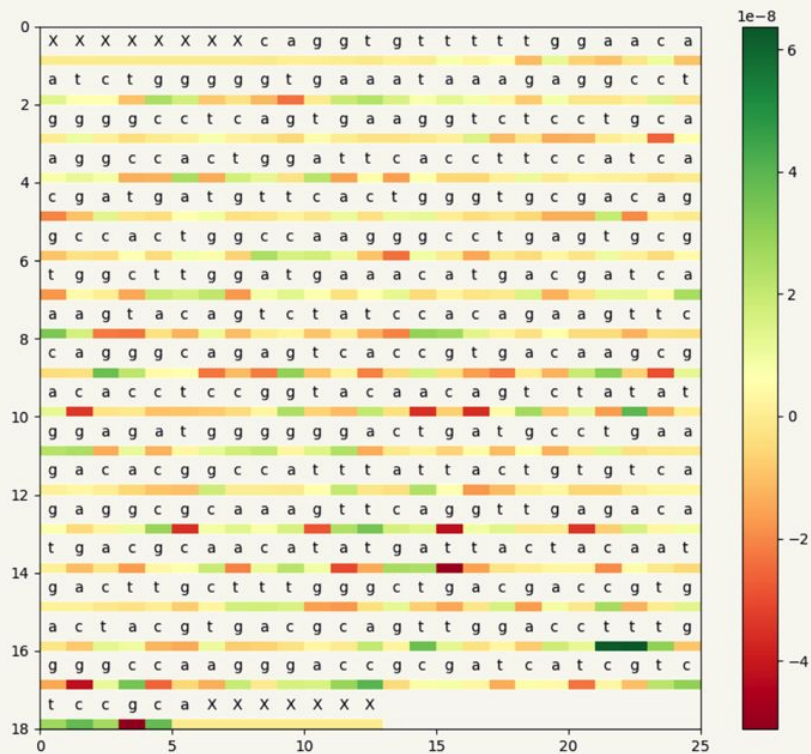


# Image recognition in Python





HIV == HIV





Thank you

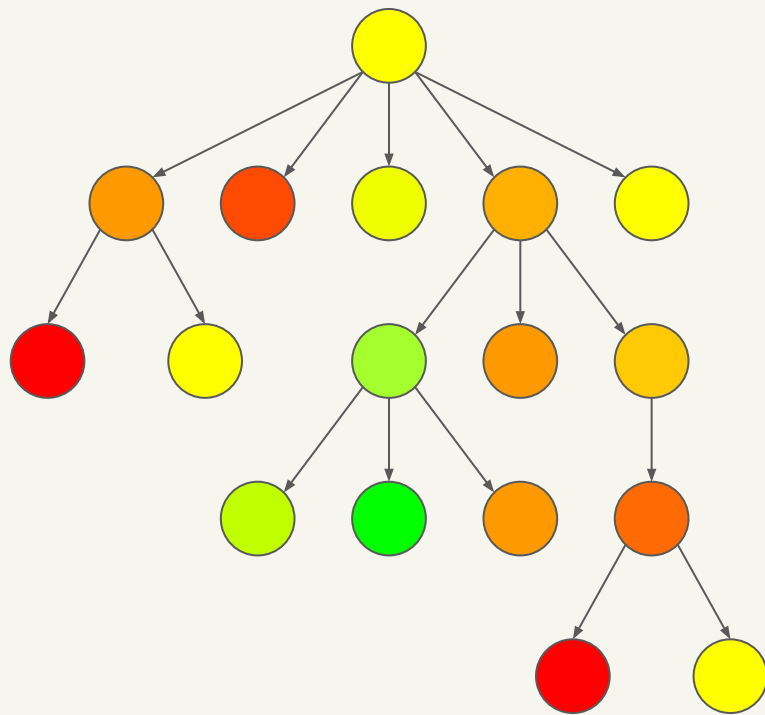
I really appreciate  
the help :)

Amazing!!



What the !\*%\$ is  
going on?!?

Thanks for  
nothing. What a  
horrible user  
experience





# Machine learning

```
procedure bubbleSort( A : list of sortable items )
  n = length(A)
  repeat
    swapped = false
    for i = 1 to n-1 inclusive do
      /* if this pair is out of order */
      if A[i-1] > A[i] then
        /* swap them and remember something changed */
        swap( A[i-1], A[i] )
        swapped = true
      end if
    end for
  until not swapped
end procedure
```





```

procedure bubbleSort( A : list of sortable items )
  n = length(A)
  repeat
    swapped = false
    for i = 1 to n-1 inclusive do
      /* if this pair is out of order */
      if A[i-1] > A[i] then
        /* swap them and remember something changed */
        swap( A[i-1], A[i] )
        swapped = true
      end if
    end for
  until not swapped
end procedure

```

```

def bubblesort(l: List) -> List:
  while True:
    swapped = False

    for i in range(len(l) - 1):
      if l[i] > l[i + 1]:
        tmp = l[i]
        l[i] = l[i + 1]
        l[i + 1] = tmp
        swapped = True

    if not swapped:
      break

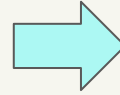
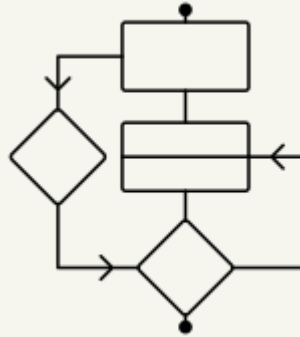
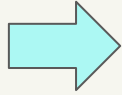
  return l

```



**Inputs:**

[3, 5, 1]  
[2]  
[2, 3, 4, 5]  
[]  
[2, 1]



**Outputs:**

[1, 3, 5]  
[2]  
[2, 3, 4, 5]  
[]  
[1, 2]

Inputs:

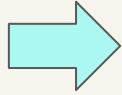
[3, 5, 1]

[2]

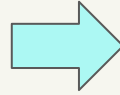
[2, 3, 4, 5]

[]

[2, 1]



?



Outputs:

[1, 3, 5]

[2]

[2, 3, 4, 5]

[]

[1, 2]

Eiendom / Bolig til salgs

527 treff i 202 annonser

Lagre søk



Kart

Sortér på

Publisert



sofienberg

Publisert

☐ Nye i dag (68)

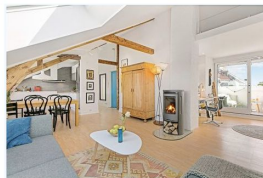
Område, by eller sted

☐ Søk etter adresse eller sted



Område

- ☐ Akershus (0)
- ☐ Aust-Agder (0)
- ☐ Buskerud (0)
- ☐ Finnmark (0)
- ☐ Hedmark (0)
- ☐ Hordaland (0)
- ☐ Møre og Romsdal (0)
- ☐ Nordland (0)



Ukens bolig

Visning i dag



Markveien 1 A, Oslo, Grünerløkka - Sofienberg

**ROLIG PÅ ØVRE GRÜNERLØKKA:** Lys og lekker 3-roms loftsleilighet med privat solrik takterrasse og felles takterrasse. -Idyllisk bakgård -Hems -

78 m² 6 200 000,-

Andel • Leilighet • 2 soverom

Schala & Partners Grünerløkka

Fellegjeld: 28 064,- • Fellesutg.: 5 226,-



Sverdrups gate 10A, Oslo, Grünerløkka - Sofienberg

**GRÜNERLØKKA - Unik 2-roms selveier** over tre plan i sjarmerende omgjort stall. Terrasse, romslig hems og herlig town-

42 m² 3 400 000,-

DNB Eiendom AS

Eier (Selveier) • Leilighet • 1 soverom

Fellegjeld: 65 000,- • Fellesutg.: 2 215,-



Visning i dag

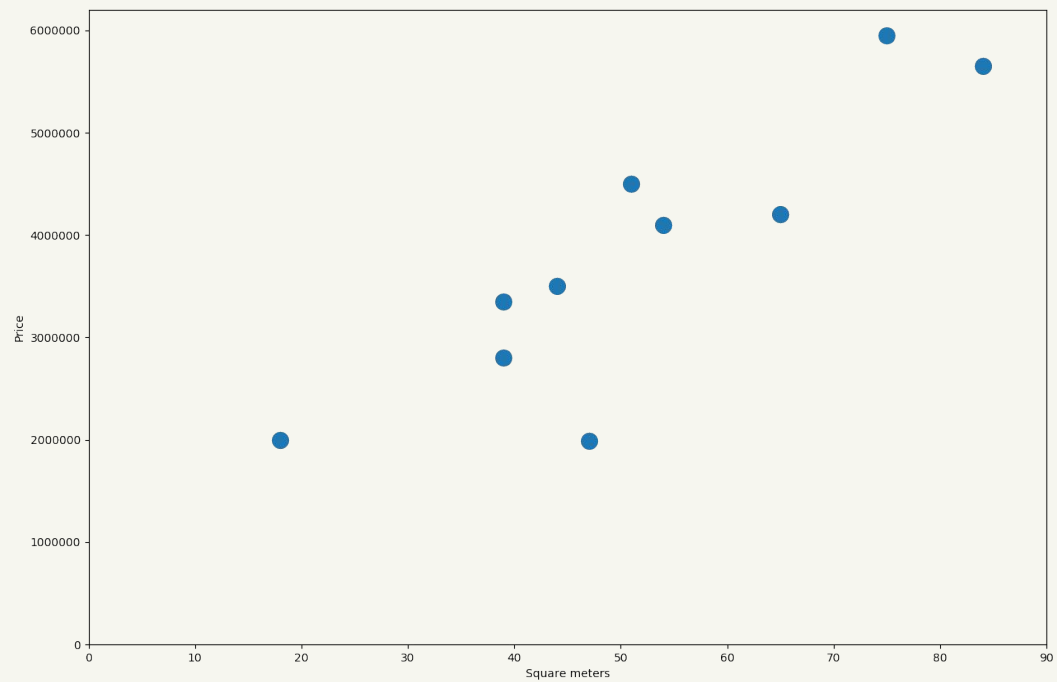


Lørenveien 57-63, leil. B2-35, Oslo

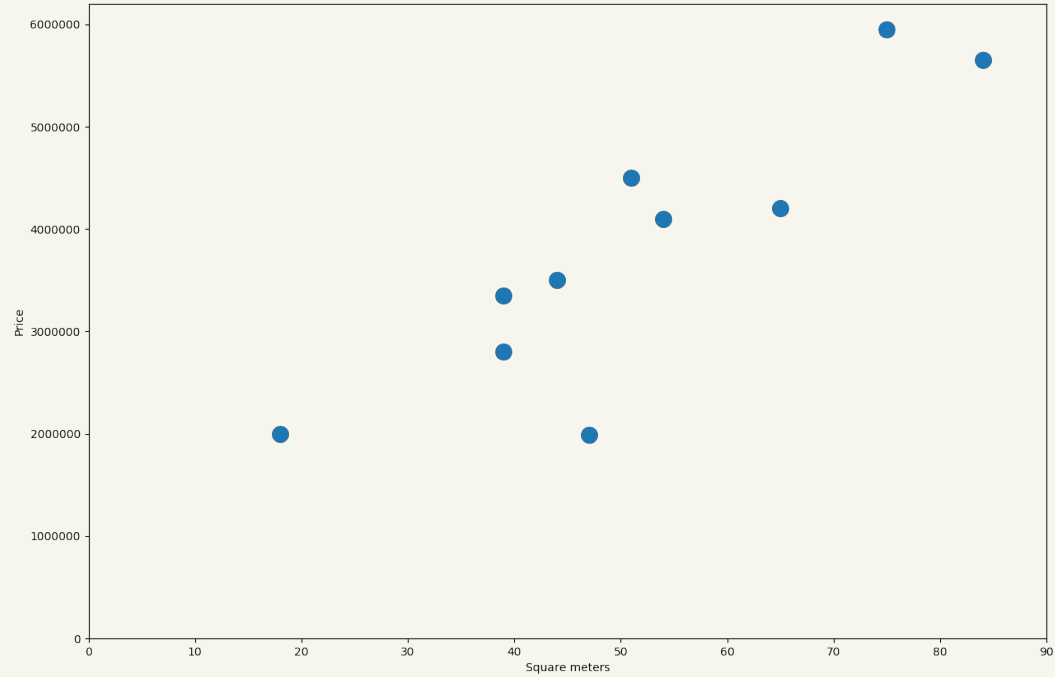
**Fastpris: Lørenporten - Ny 2R m/ca. 15 kvm vestvendt terrasse. 31.12.2020 tas det sikte på overtakelse.**

<b>m<sup>2</sup></b>	<b>Price</b>
51	4 500 000
84	5 650 000
44	3 500 000
47	1 990 000
75	5 950 000
39	3 350 000
18	2 000 000
65	4 200 000
54	4 100 000
39	2 800 000

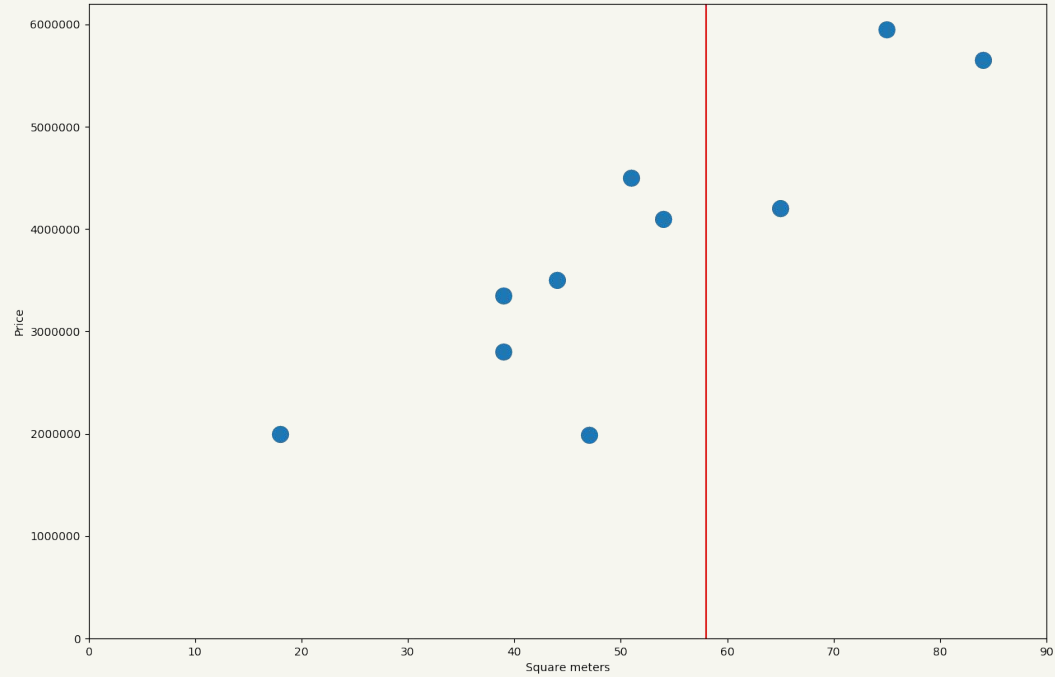




$$\text{price} = f(\text{m}^2)$$

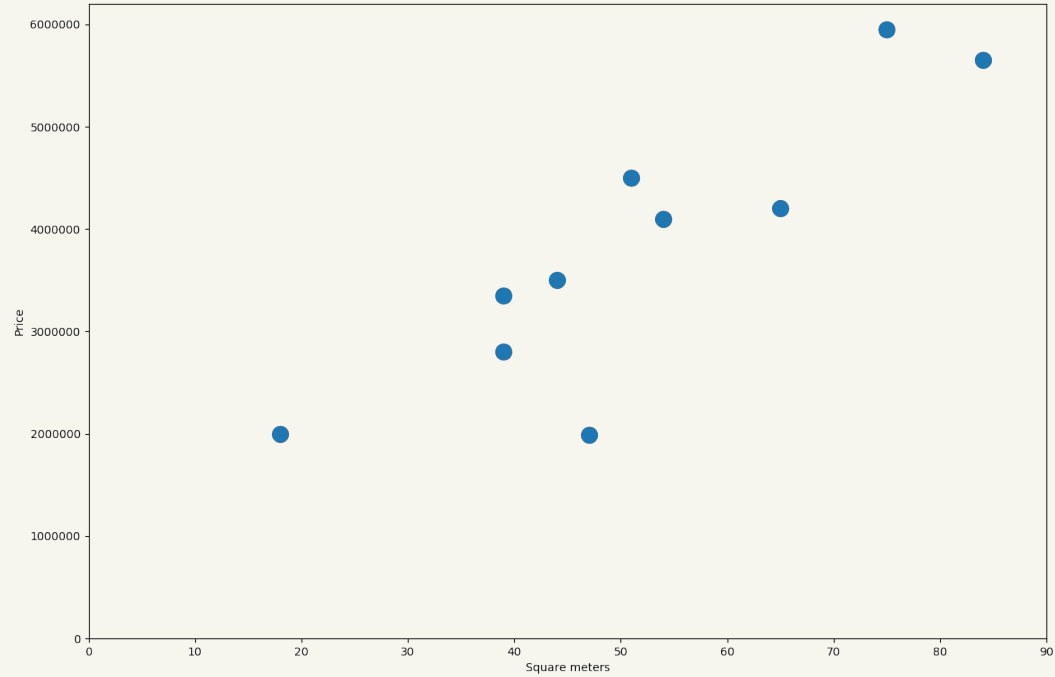


$$\text{price} = f(58)$$

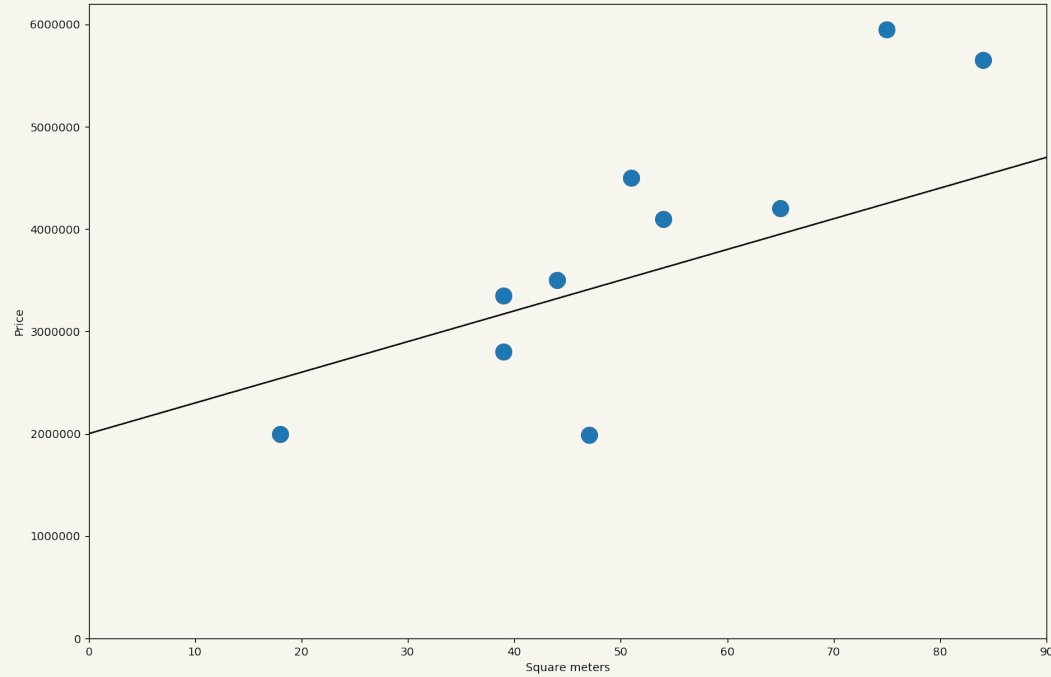




$$\text{price} = a(m^2) + b$$



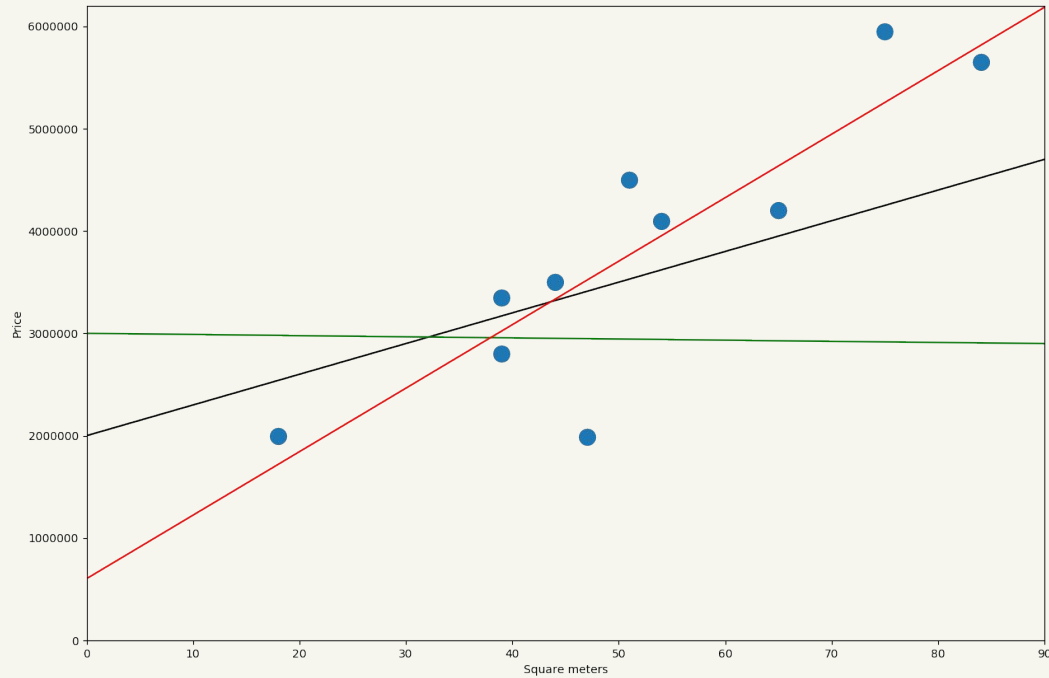
$$\text{price} = 30000(\text{m}^2) + 2000000$$



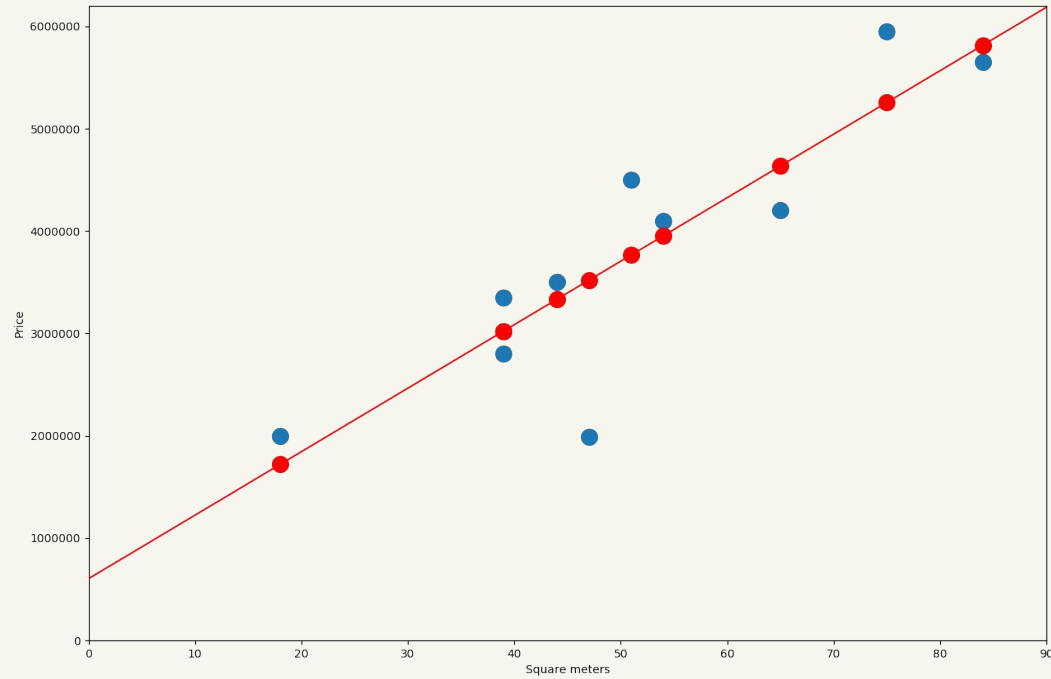
$$\text{price} = 10000(\text{m}^2) + 3000000$$

$$\text{price} = 30000(\text{m}^2) + 2000000$$

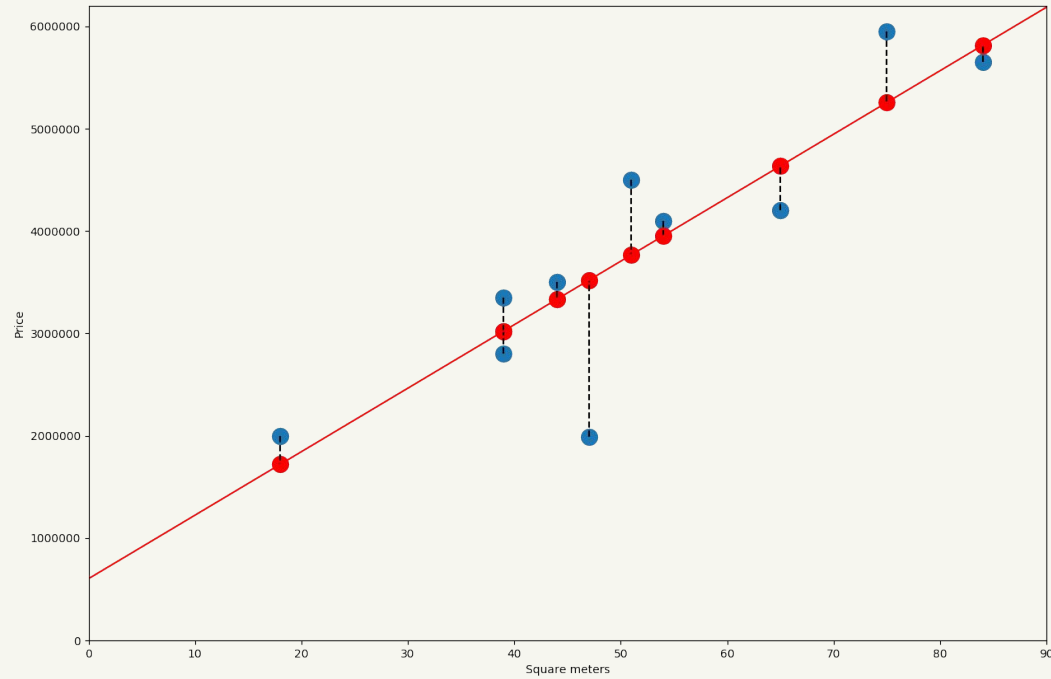
$$\text{price} = 62044(\text{m}^2) + 602535$$



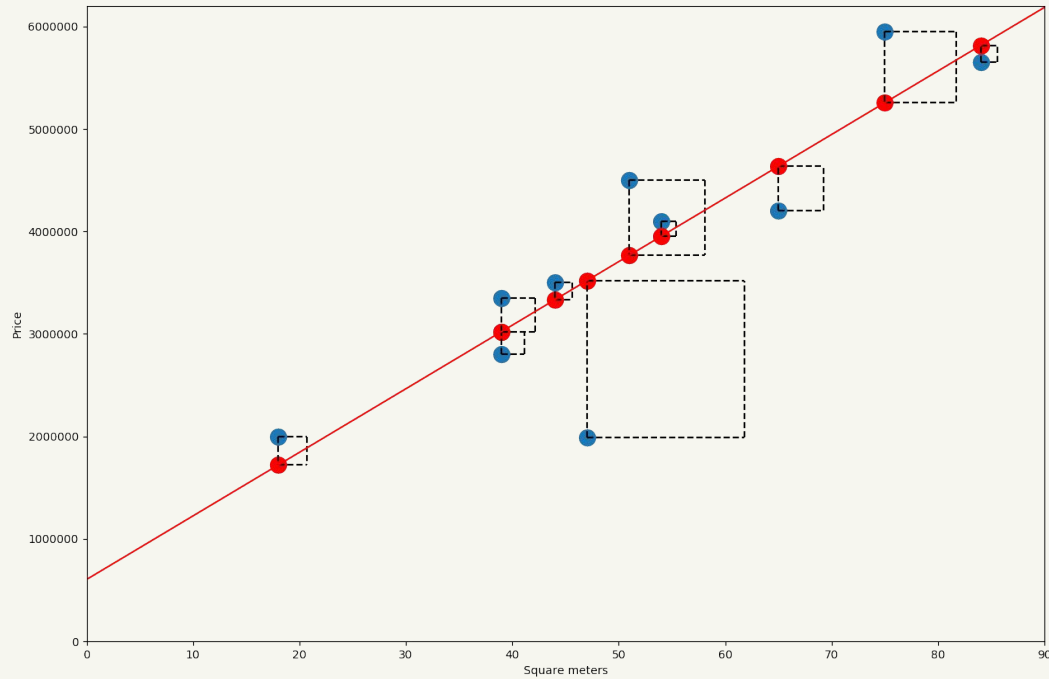
$$\text{price} = 62044(\text{m}^2) + 602535$$



$$\text{price} = 62044(\text{m}^2) + 602535$$



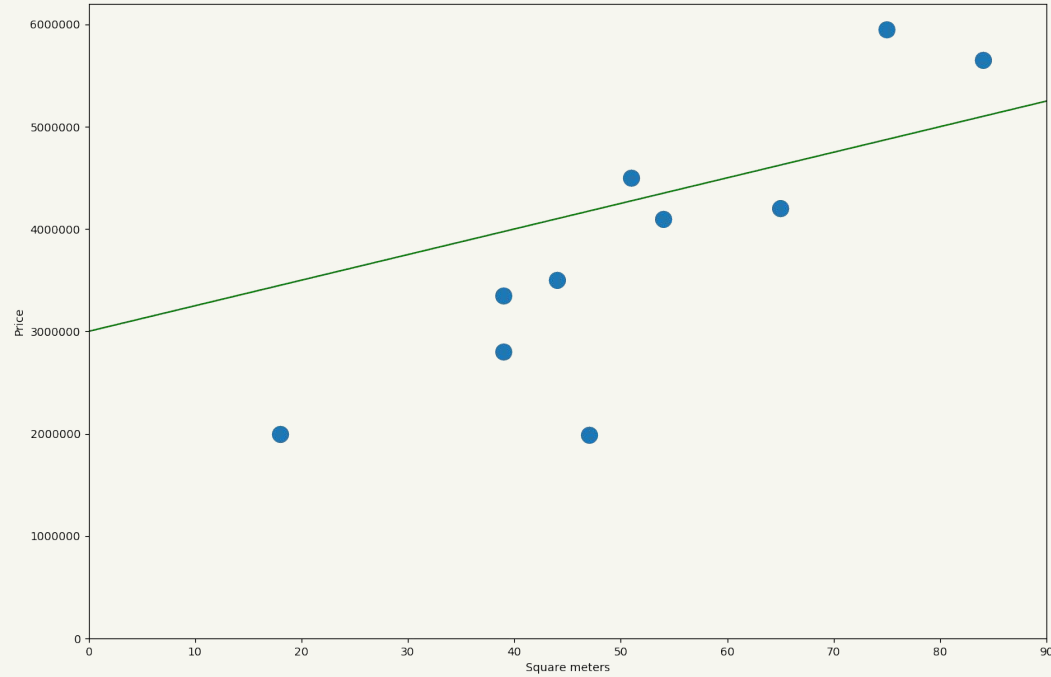
$$\text{price} = 62044(\text{m}^2) + 602535$$



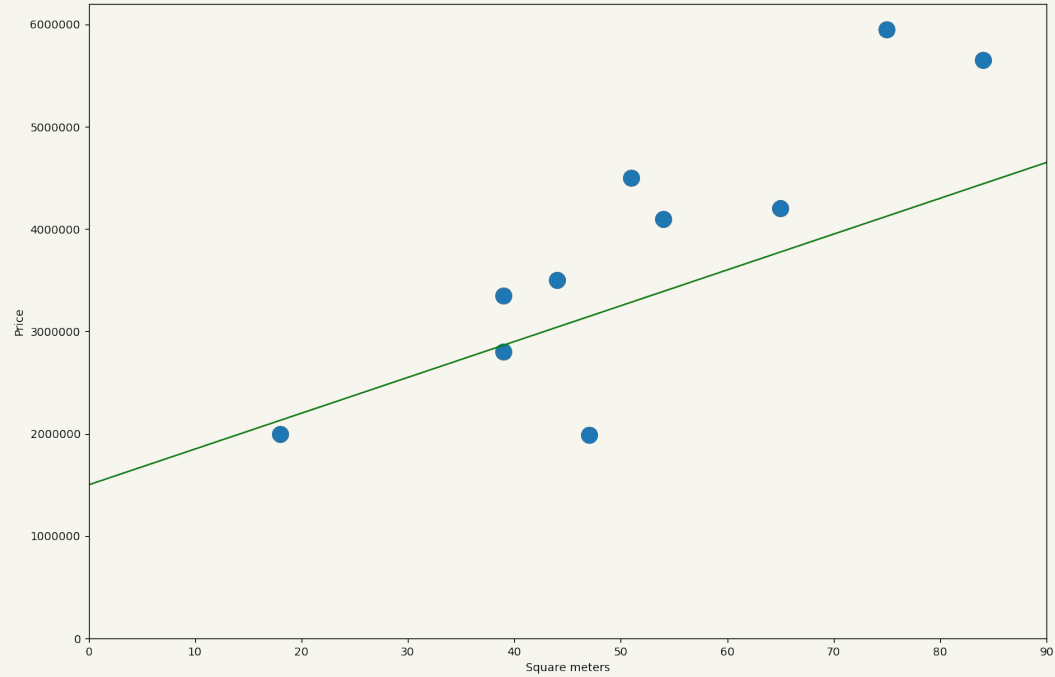
$$mse(f) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2$$



$$\text{price} = 25000(\text{m}^2) + 3000000$$

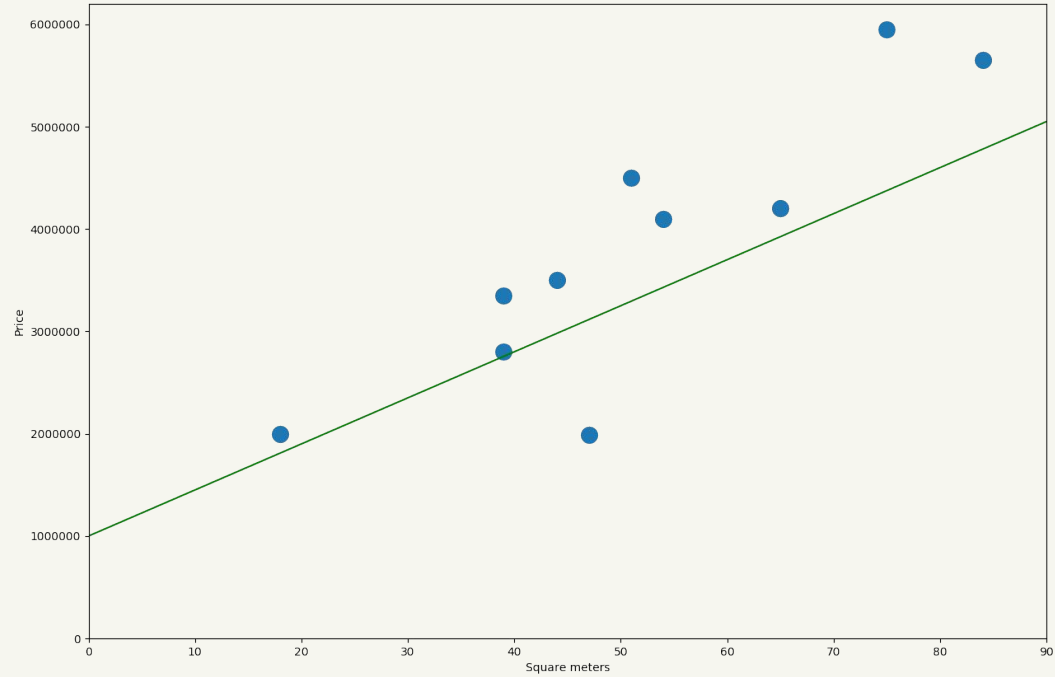


$$\text{price} = 35000(\text{m}^2) + 1500000$$

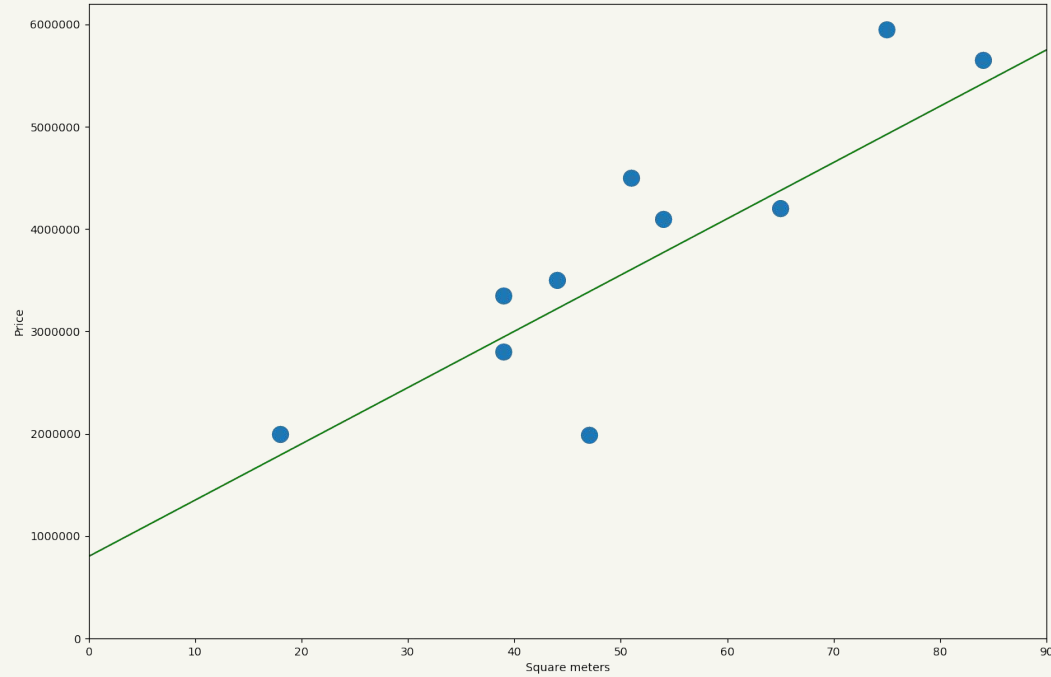




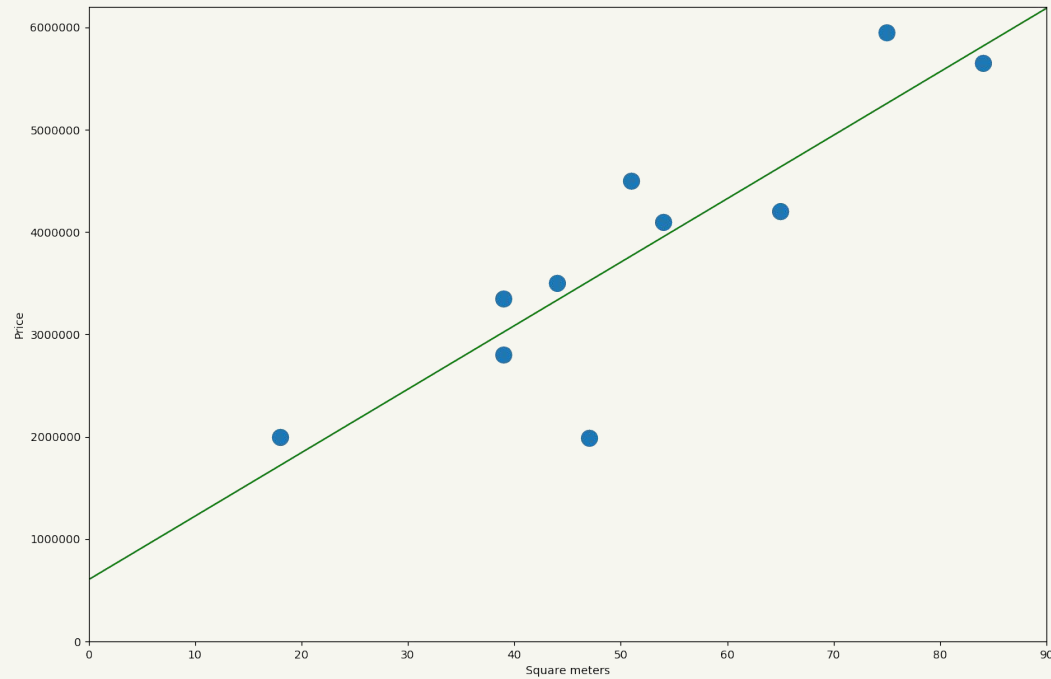
$$\text{price} = 45000(\text{m}^2) + 1000000$$



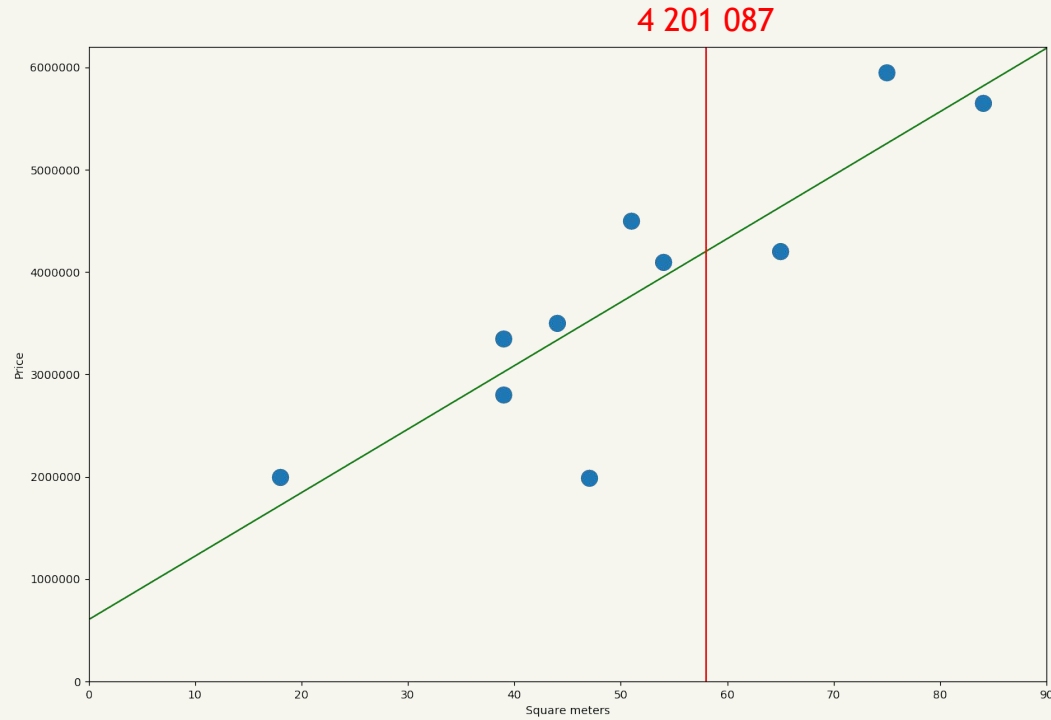
$$\text{price} = 55000(\text{m}^2) + 800000$$



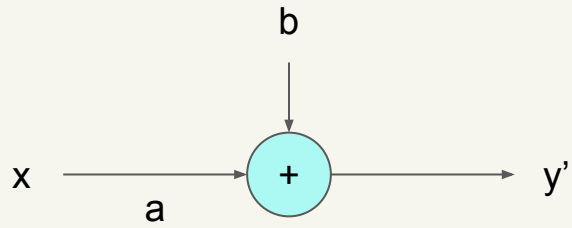
$$\text{price} = 62044(\text{m}^2) + 602535$$



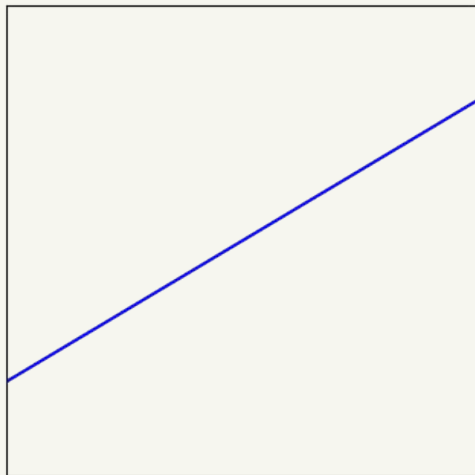
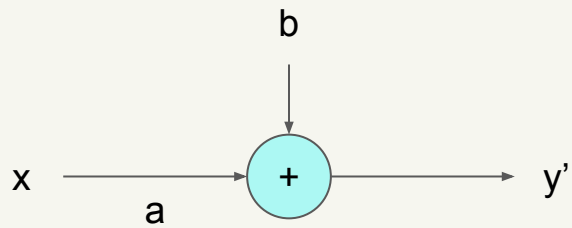
$$\text{price} = 62044(\text{m}^2) + 602535$$

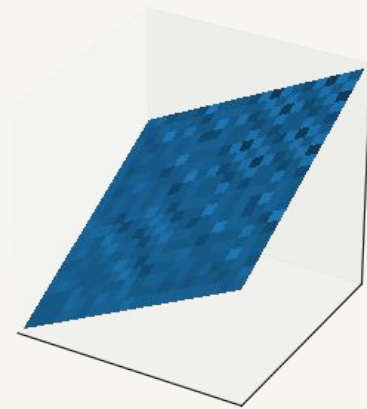
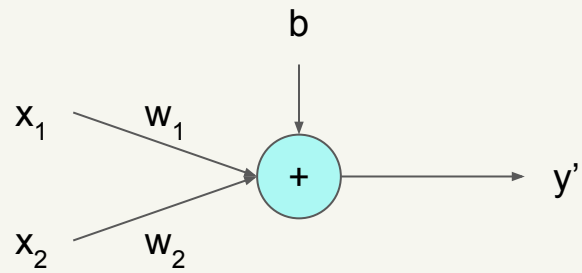


# Image recognition

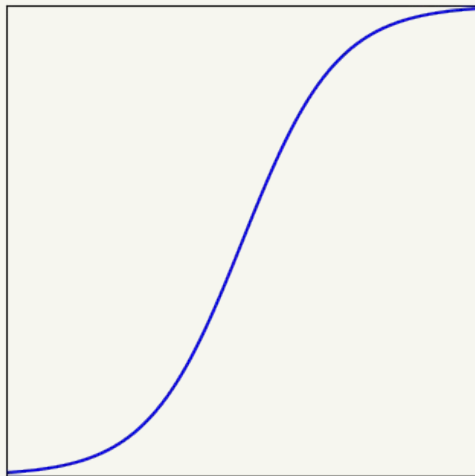
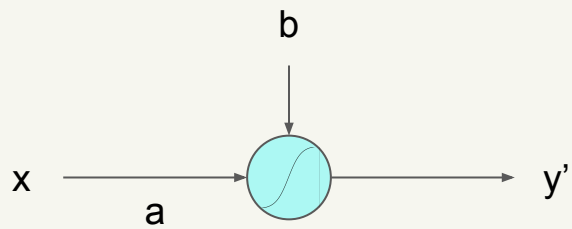


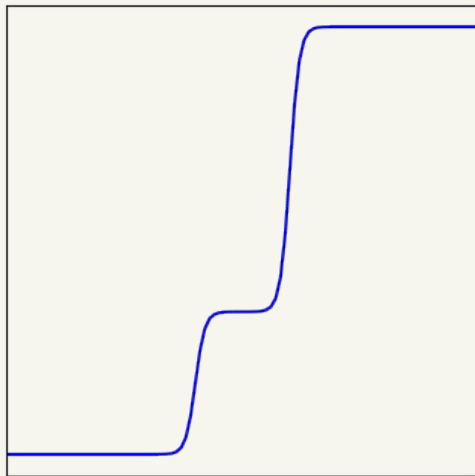
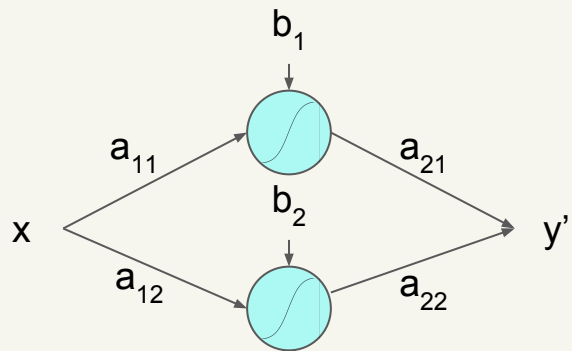
$$y' = ax + b$$

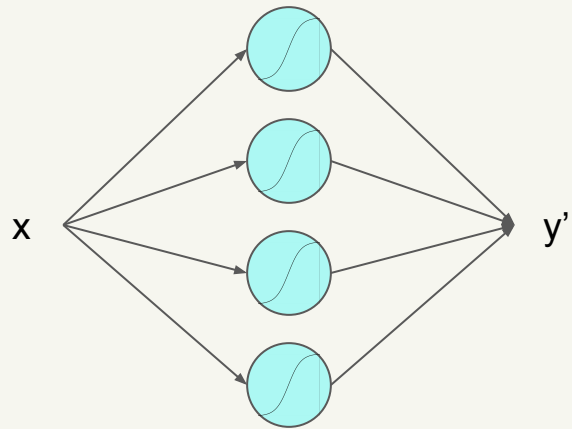


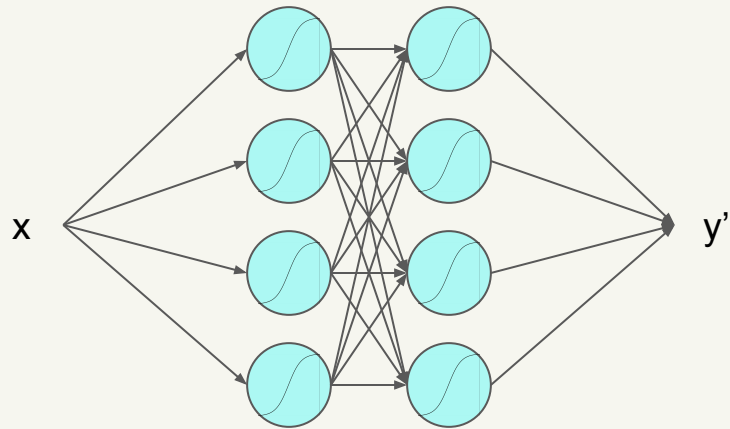


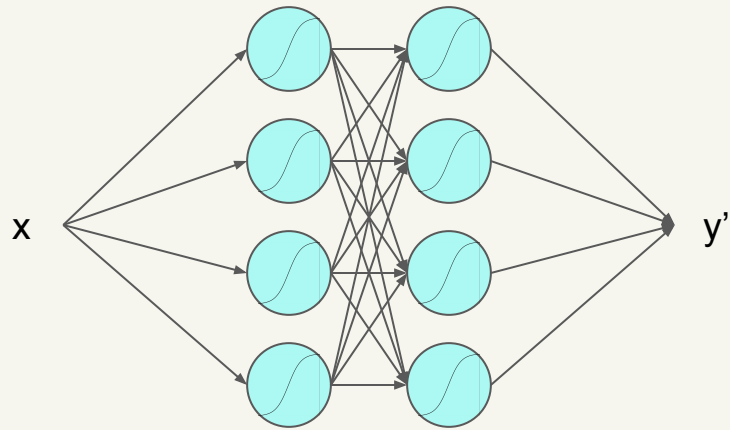




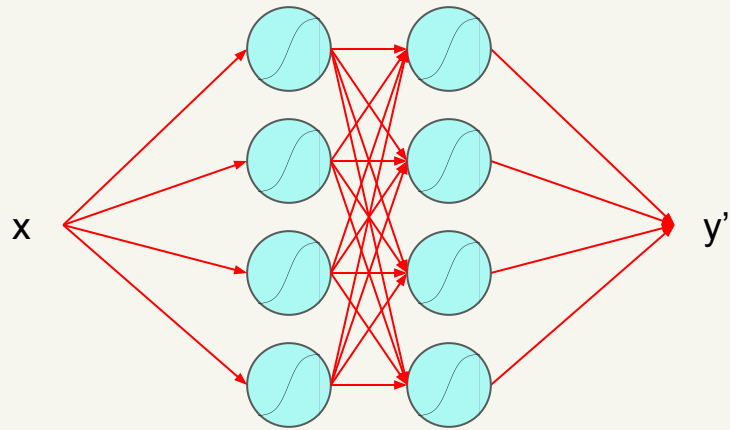




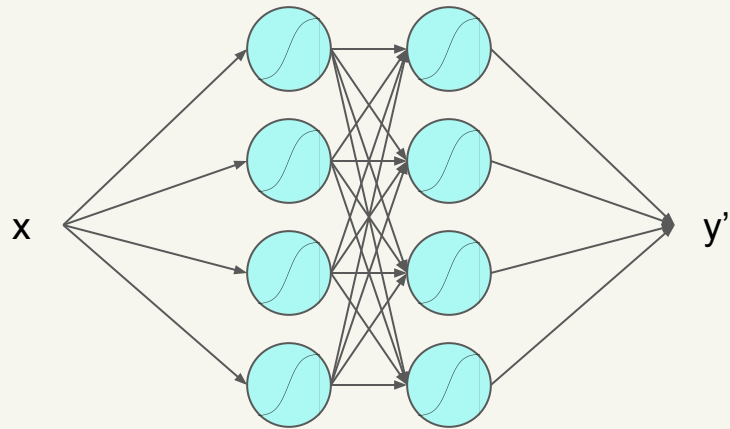


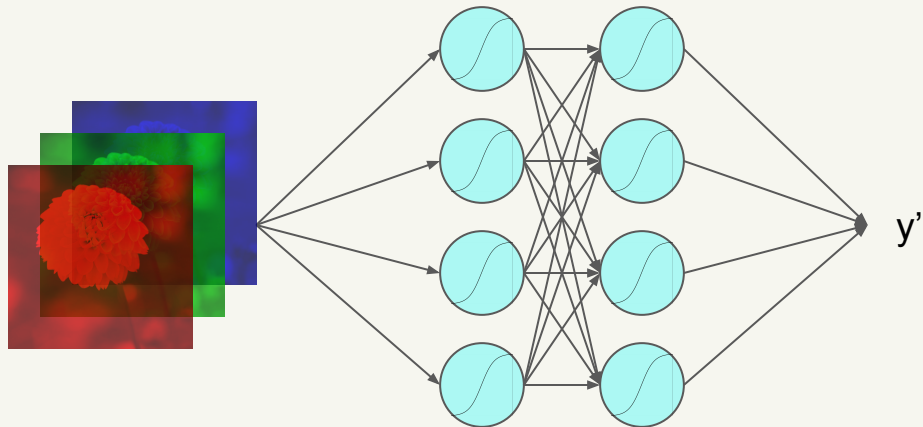


$\text{loss}(y, y')$

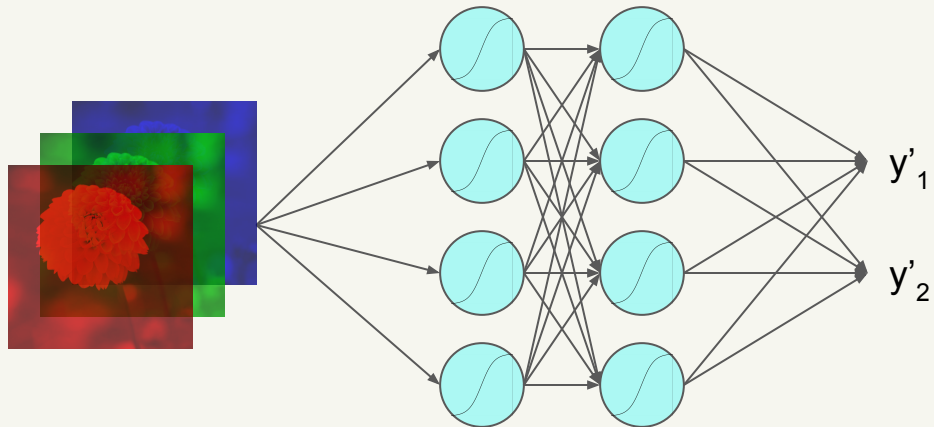


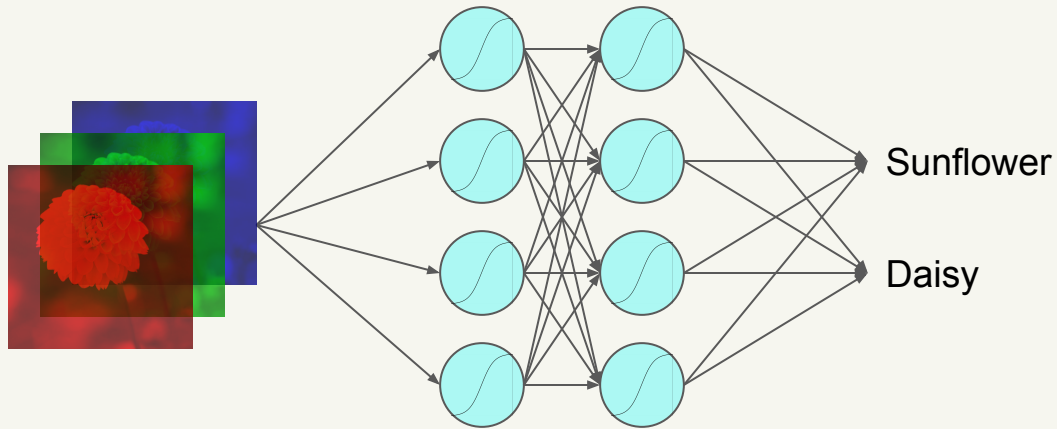
$\text{loss}(y, y')$





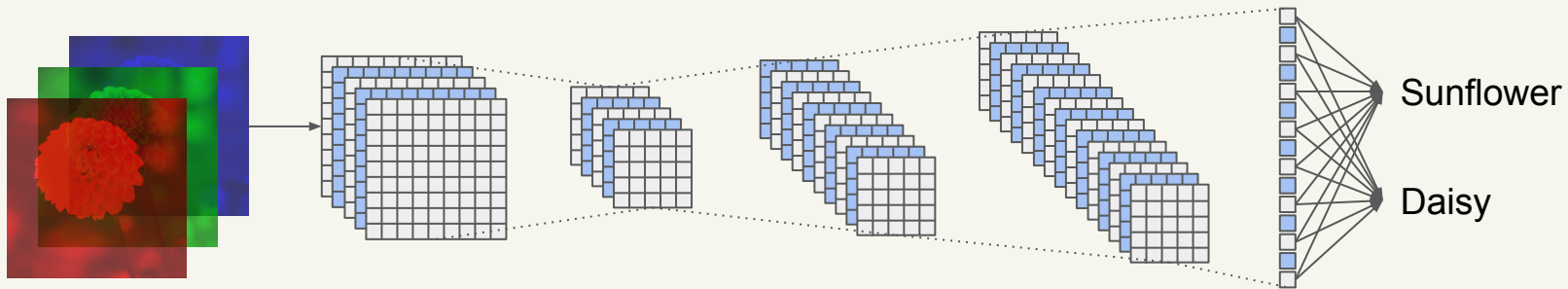


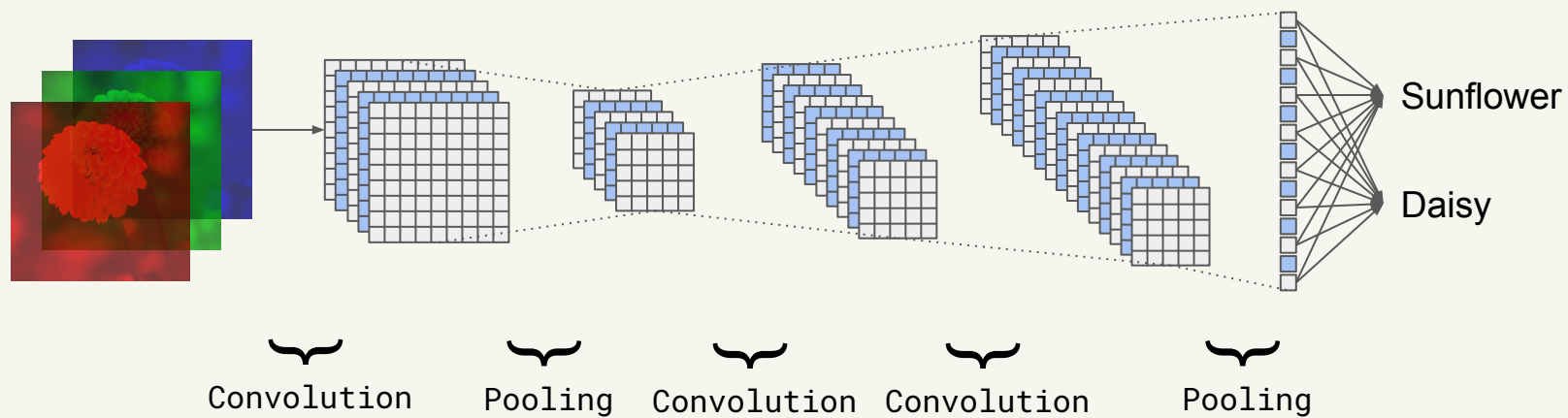


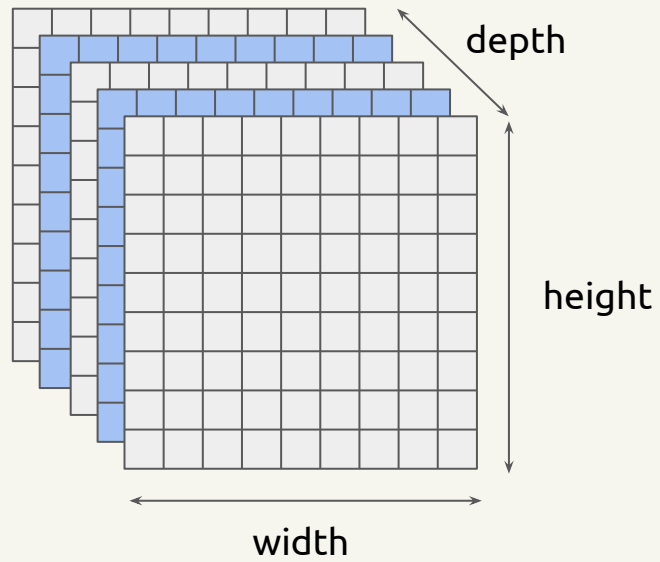


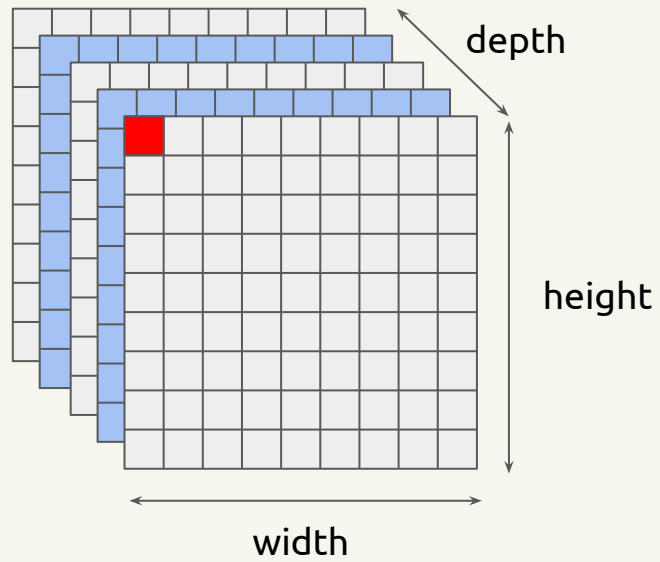
# Convolutional Neural Nets

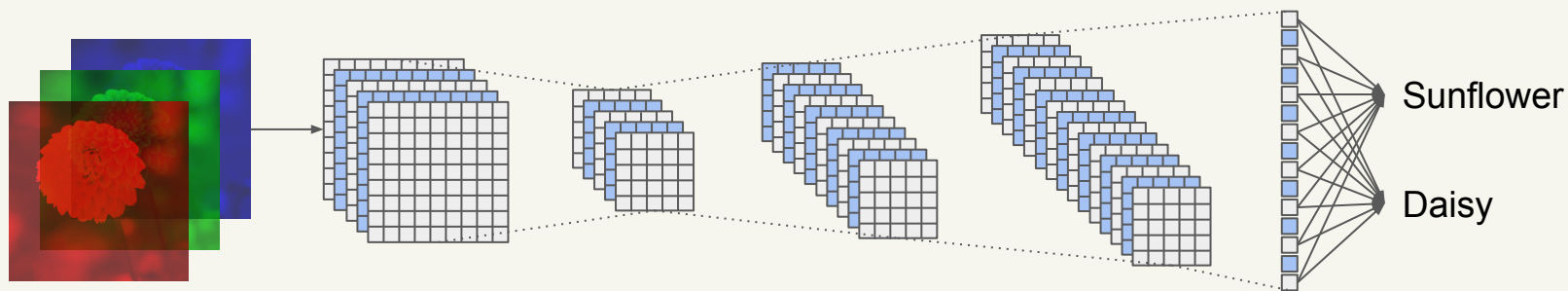
A faint, stylized diagram of a Convolutional Neural Network (CNN) architecture is visible in the background. It shows a sequence of layers: an input layer on the left, followed by several convolutional layers with feature maps of varying sizes, and finally a fully connected layer on the right. The diagram uses various shapes like rectangles and circles to represent different components and their connections.



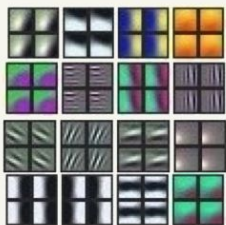






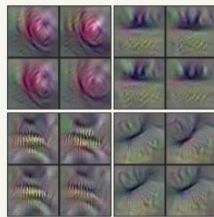


Convolution

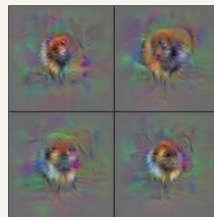


Pooling

Convolution

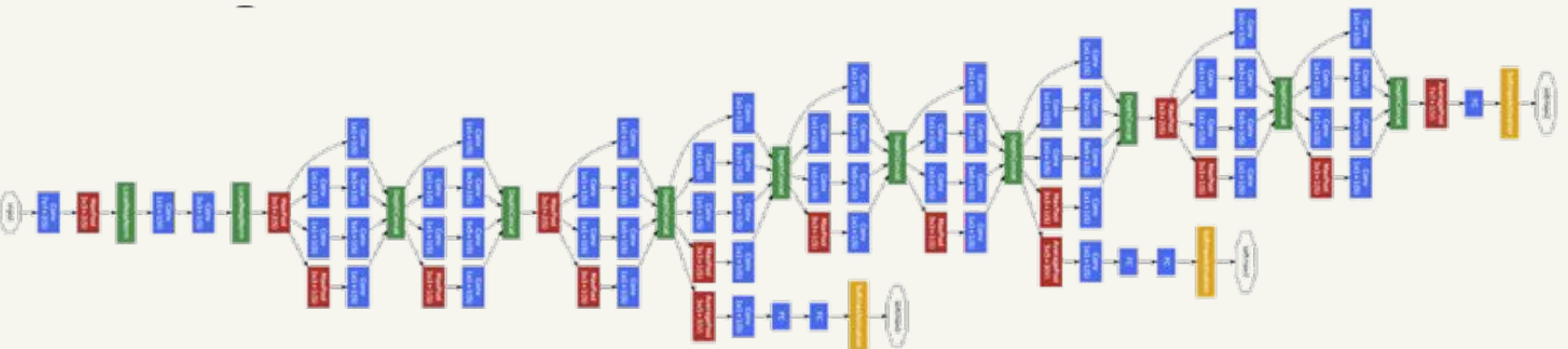


Convolution



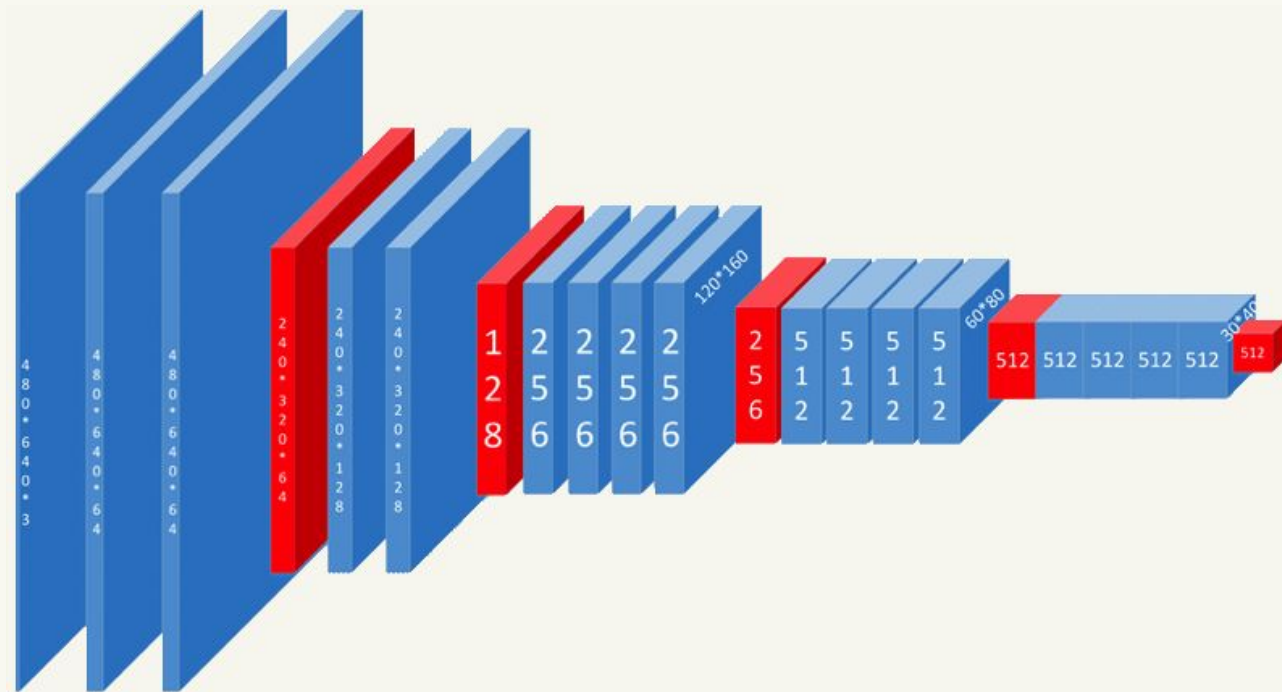
Pooling





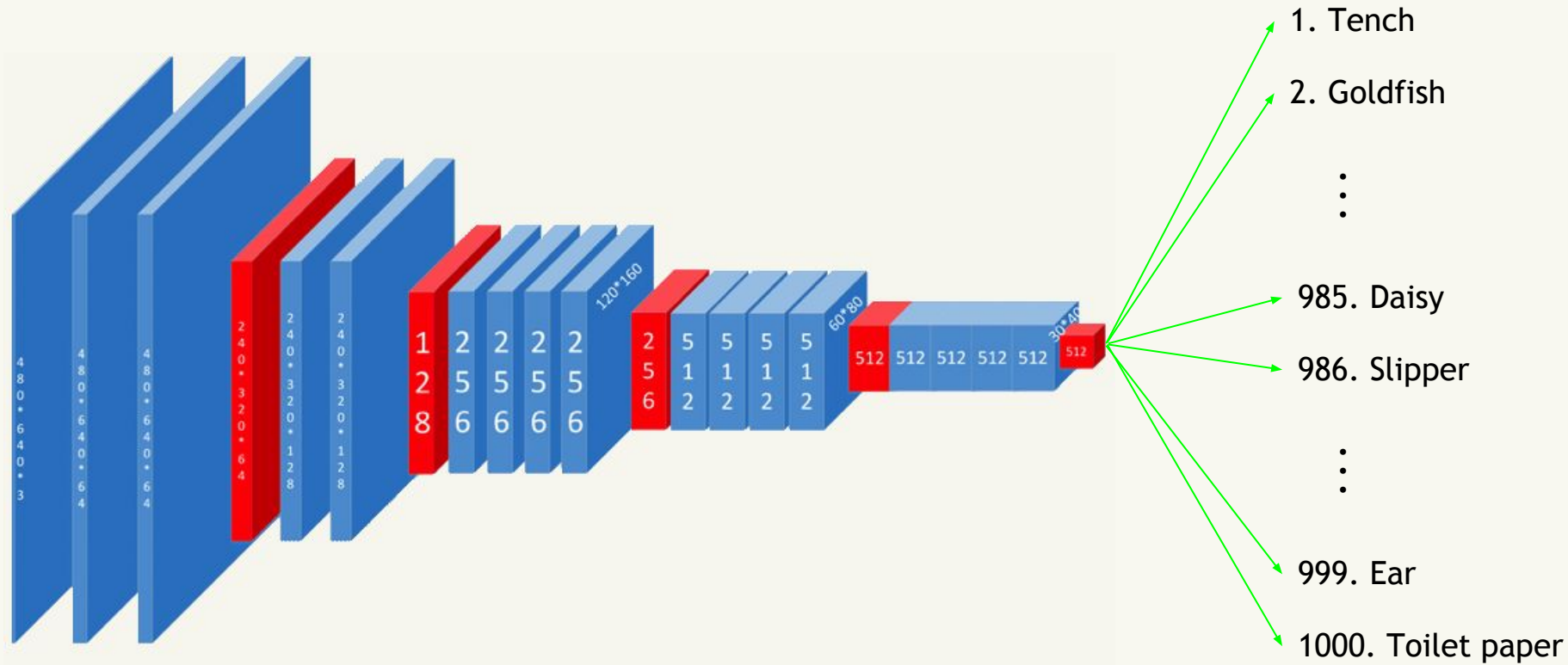


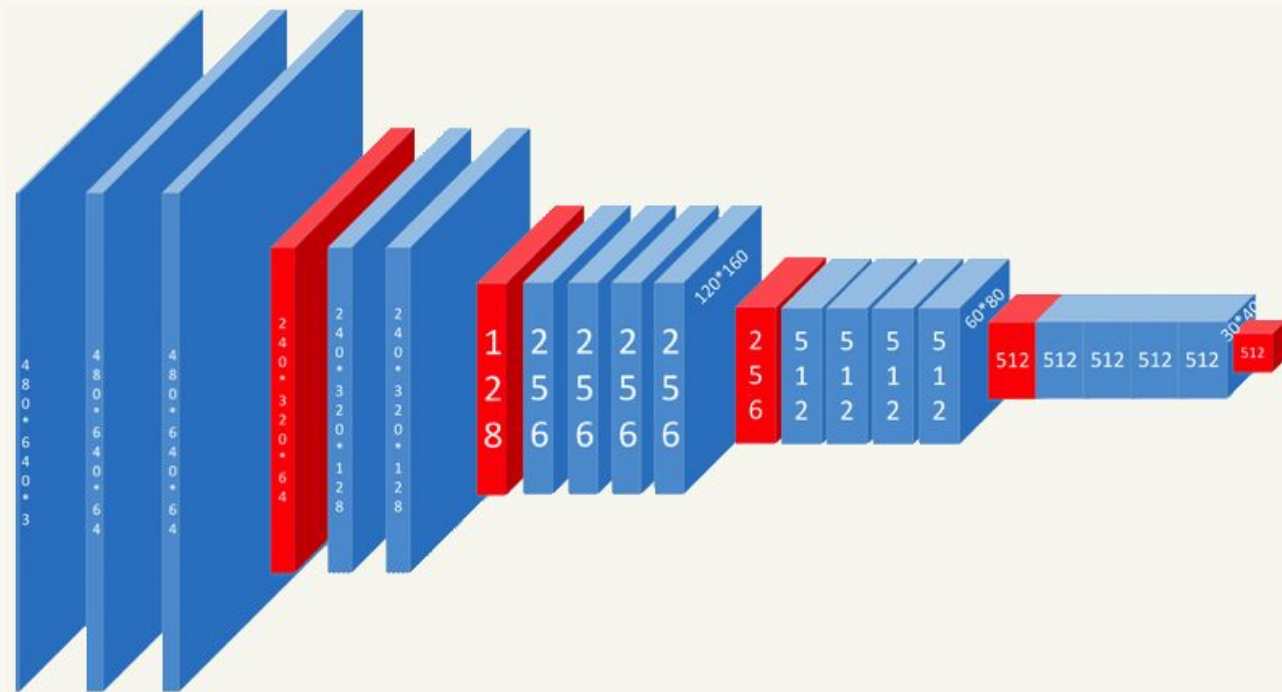
# Transfer learning



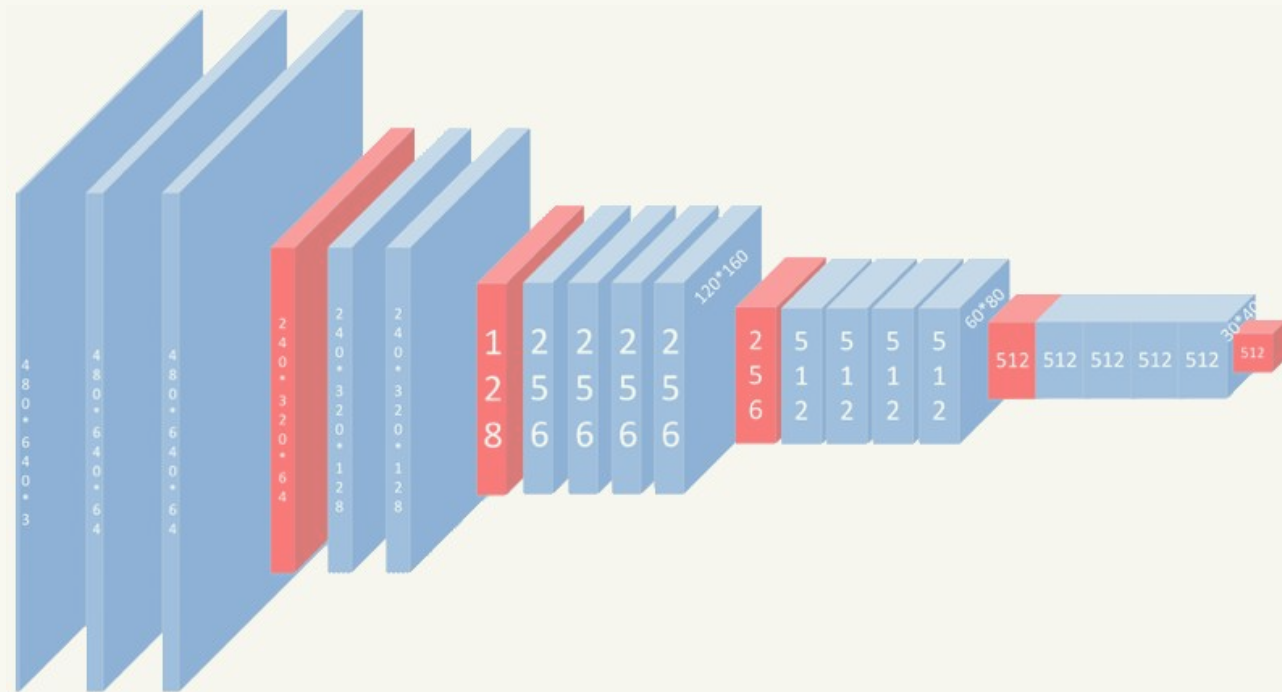






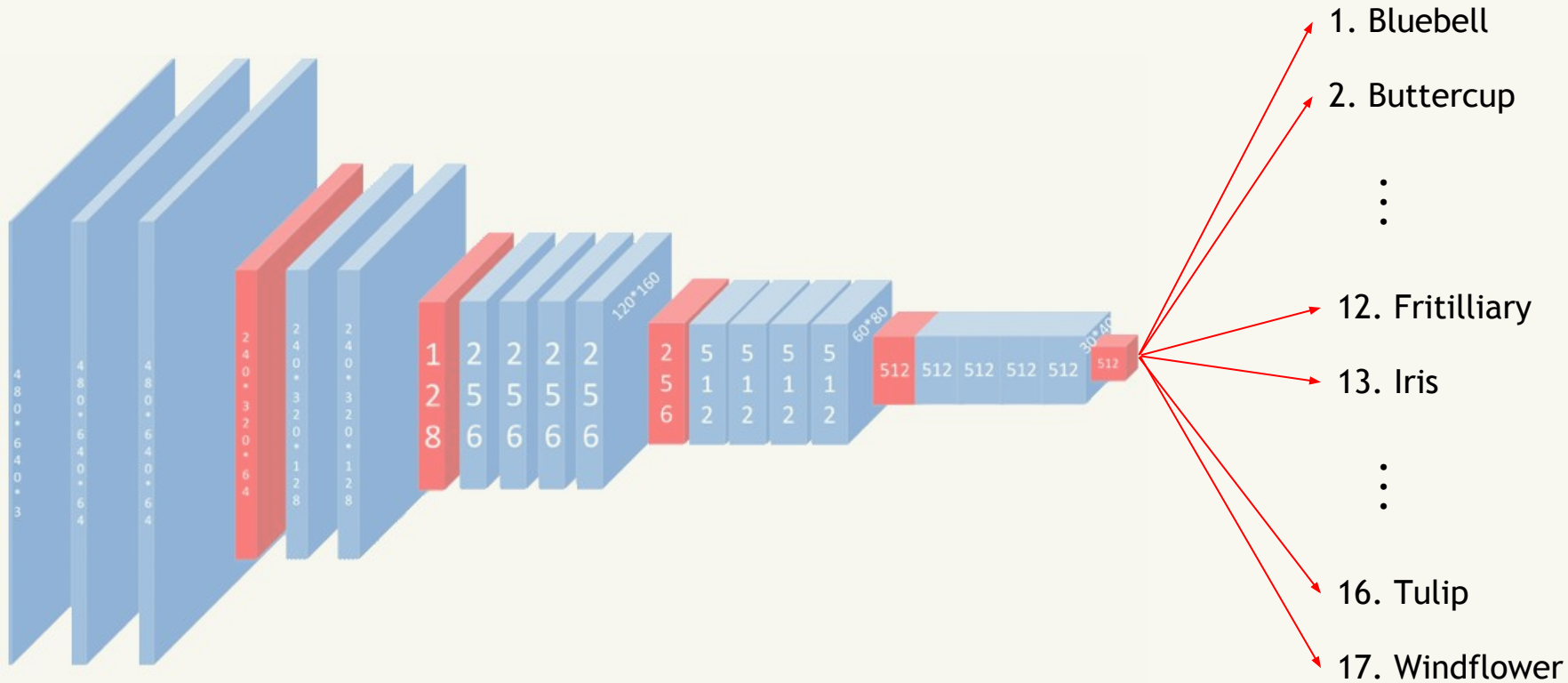




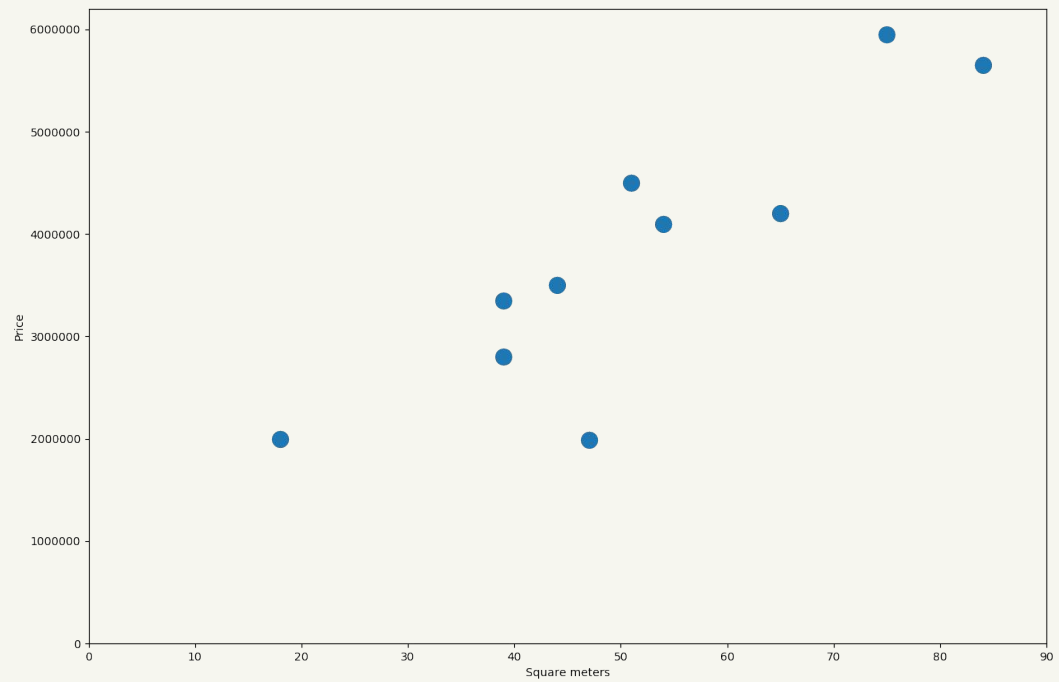


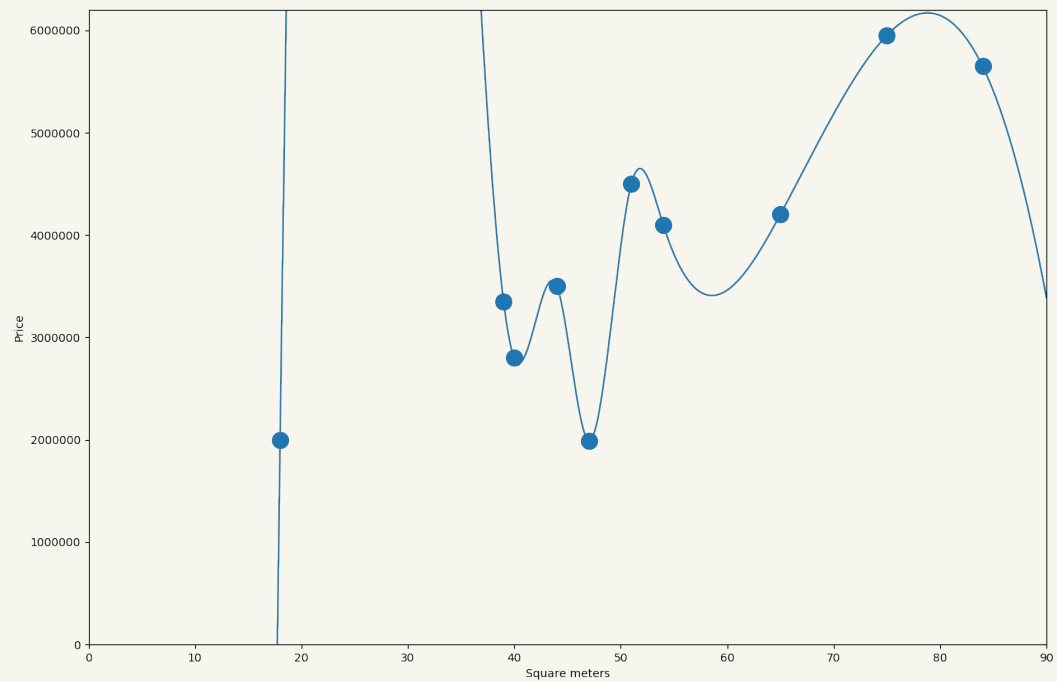


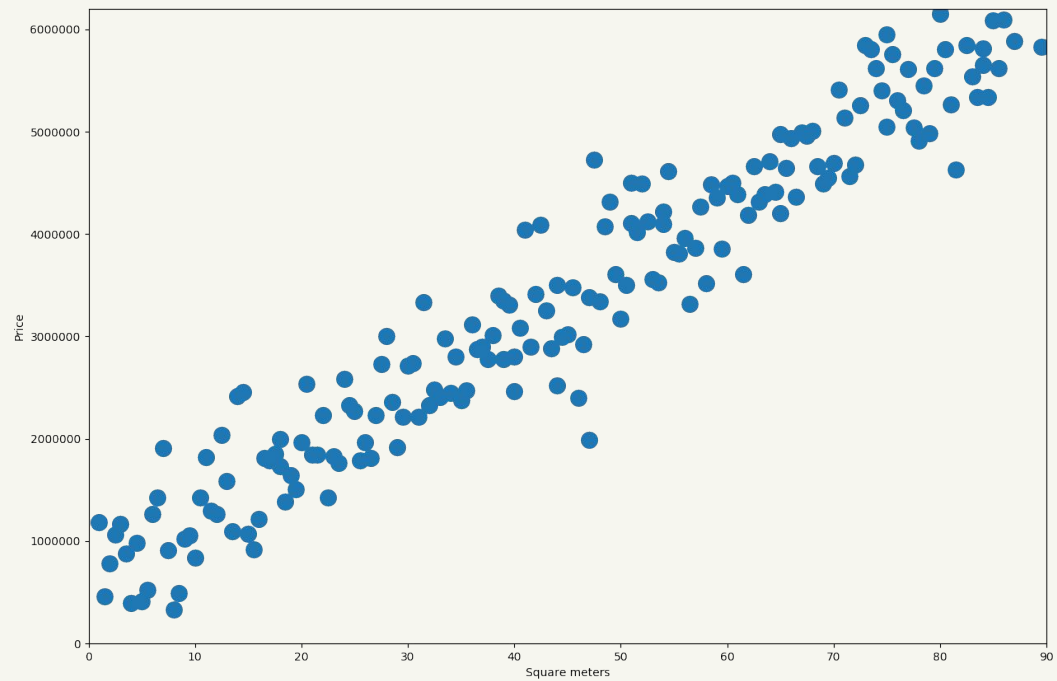


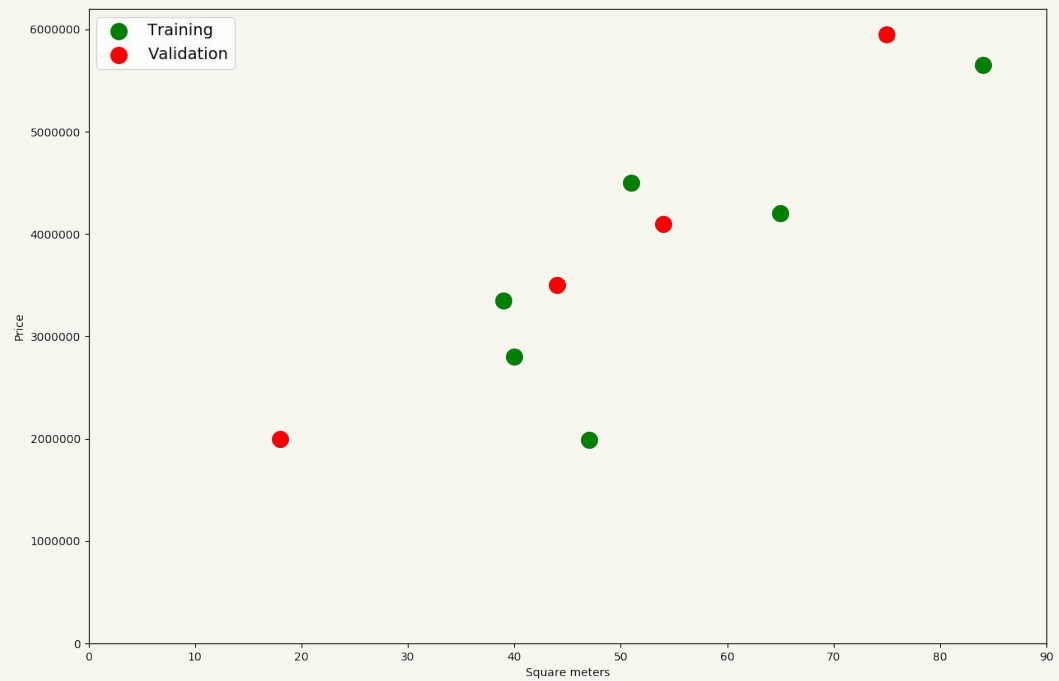


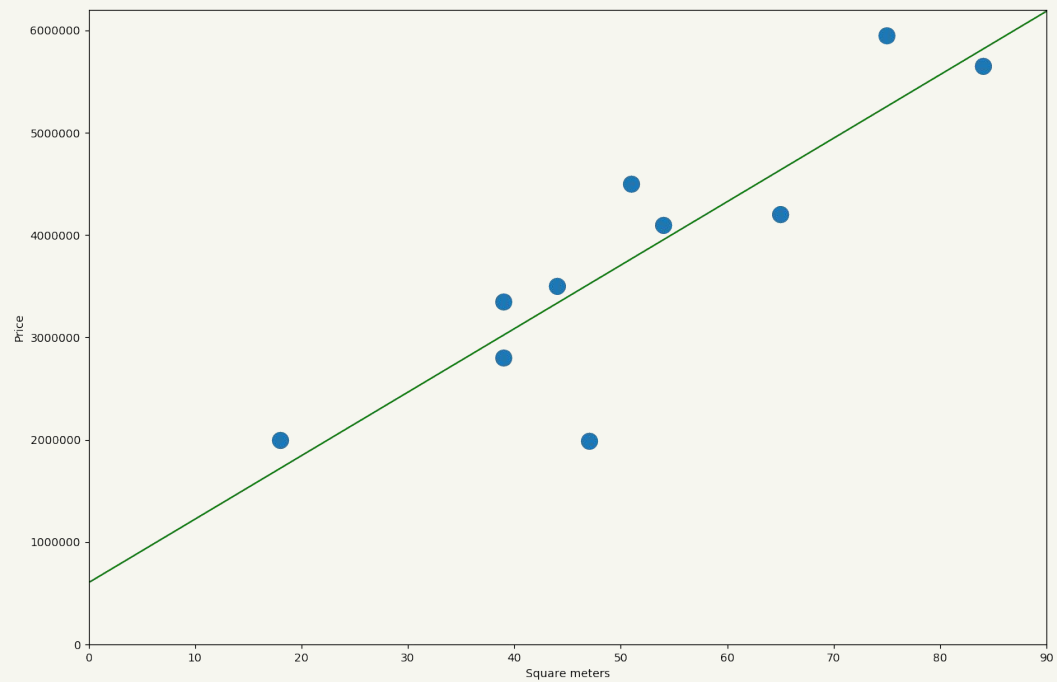
# Overfitting











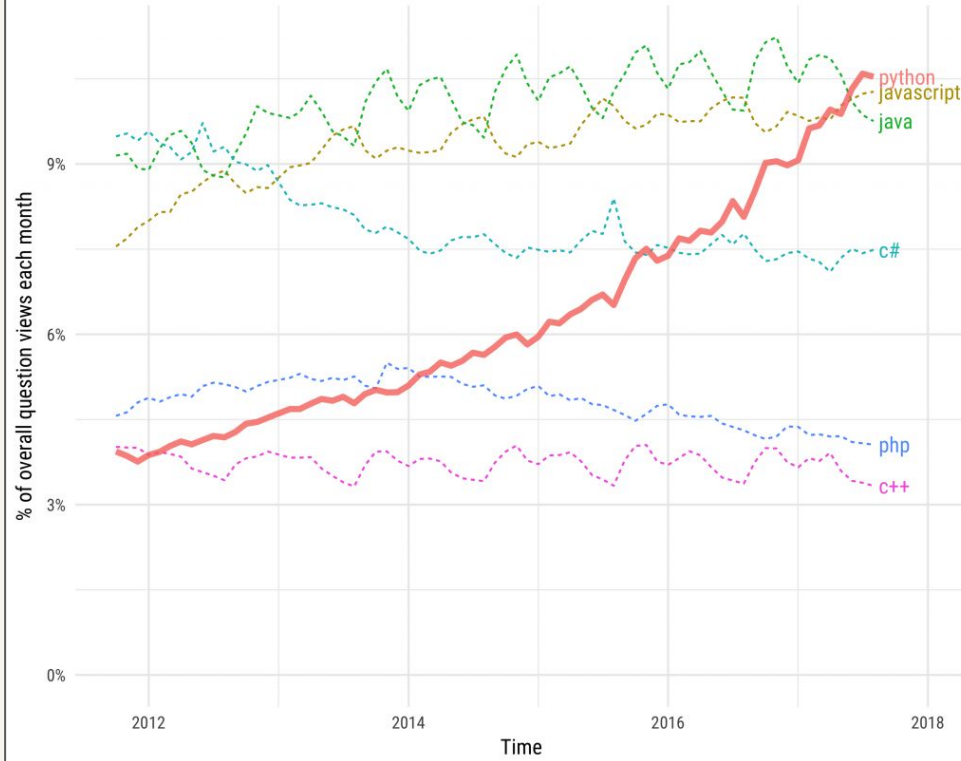


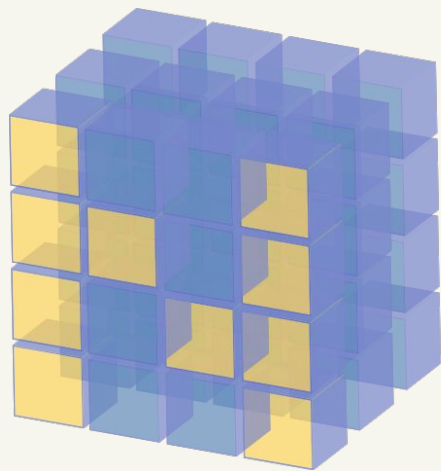
# Flower classification



## Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries





NumPy



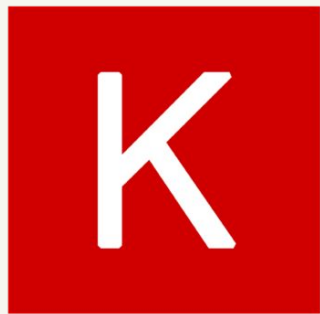
TensorFlow



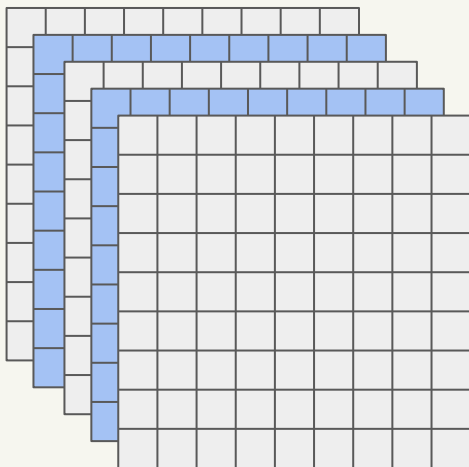
$$y = ax + b$$

```
x = tf.placeholder(  
    dtype=tf.float32,  
    shape=(4, 4)  
)  
a = tf.Variable(  
    initial_value=tf.zeros((4, 4)),  
    dtype=tf.float32,  
    trainable=True  
)  
b = tf.Variable(  
    initial_value=tf.zeros((4, 4)),  
    dtype=tf.float32,  
    trainable=True  
)  
  
y = tf.add(tf.matmul(x, a), b)
```





Keras



```
BATCH_SIZE = 4
IMAGE_SIZE = (256, 256, 3)

inputs = tf.placeholder(
    shape=(BATCH_SIZE,) + IMAGE_SIZE,
    dtype=tf.float32
)

weight_shape = tf.stack([5, 5, 3, 16])
weight_initializer = tf.random_normal(
    shape=weight_shape,
    stddev=.03
)
weights = tf.Variable(
    weight_initializer,
    trainable=True,
)

bias_initializer = tf.zeros(16)
bias = tf.Variable(
    bias_initializer,
    trainable=True
)

conv = tf.nn.conv2d(inputs, weights,
    strides=[1, 1, 1, 1],
    padding='SAME'
)
conv = tf.nn.bias_add(conv, bias)
```

```
IMAGE_SIZE = (256, 256, 3)

inputs = Inputs(shape=IMAGE_SIZE)
conv = Conv2D(kernels=16, filters=(3, 3))(inputs)
```





```
from keras.applications.vgg19 import VGG19

model = VGG19()
```

```
from keras.applications.vgg19 import VGG19

model = ... # Create a model

model.summary() # Print the structure of the model

model.fit() # Train the model

model.evaluate() # Evaluate performance on a validation set

model.predict() # Run predictions on new data

model.save() # Save the model to file

model.load() # Load the model into memory
```



Serving  
images

1

Preprocessing

3

Augmentations

6

Core  
model

2

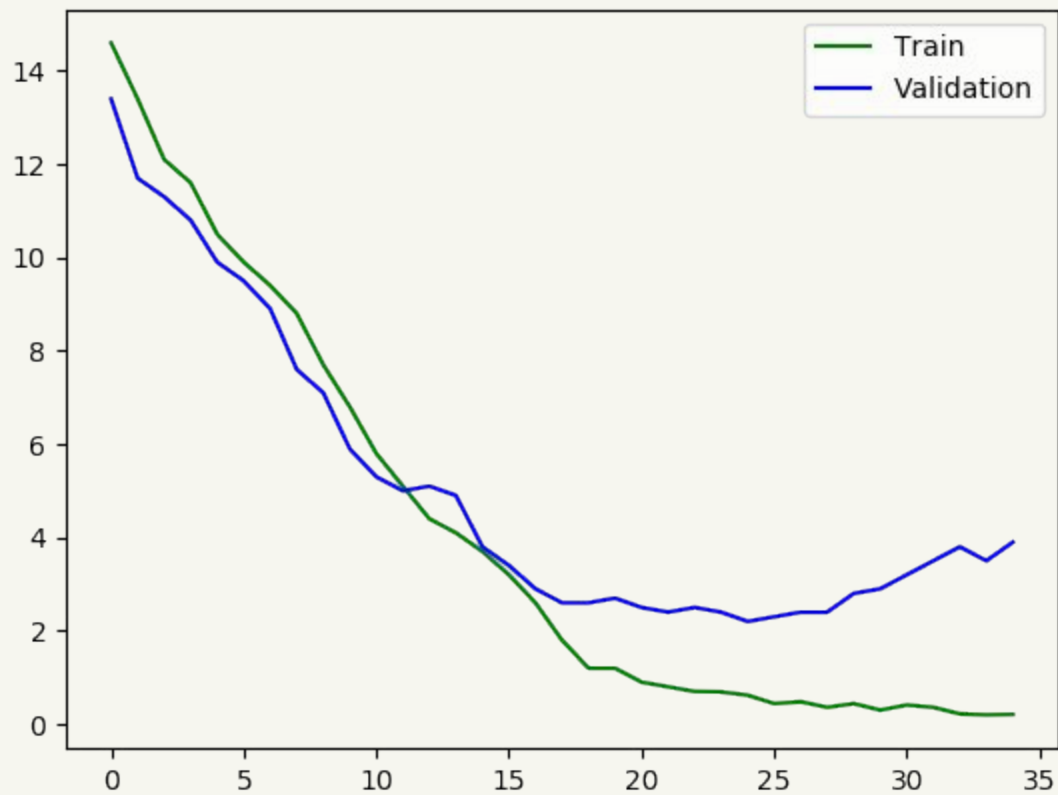
Specific  
model

4/5



<u>Labels:</u>	<u>Indexes:</u>	<u>Onehot encoded:</u>
'bluebell'	0	[1, 0, 0, 0, 0, 0]
'buttercup'	1	[0, 1, 0, 0, 0, 0]
'colts_foot'	2	[0, 0, 1, 0, 0, 0]
'cowslip'	3	[0, 0, 0, 1, 0, 0]
'crocus'	4	[0, 0, 0, 0, 1, 0]
'daffodil'	5	[0, 0, 0, 0, 0, 1]







esten@epimed.ai