Lecture 2
Data Labeling

Xu Yuan
University of Louisiana at Lafayette

Machine Learning ~ Training Framework

Dog

Monkey

Cat

Cat

Training Data

A set of functions (models) $f_1, f_2, \ldots$

Goodness of function f

Pick the "best" function f^*

Trained Model

Machine Learning ~ Testing Framework

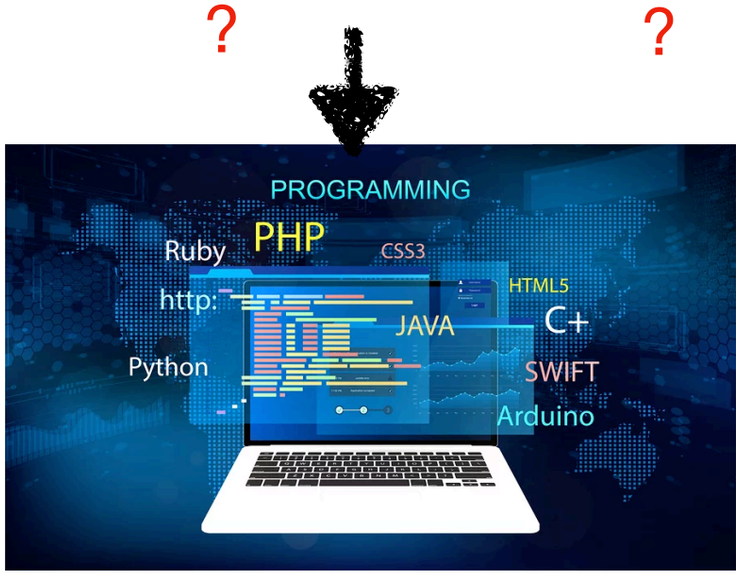
?

?

?

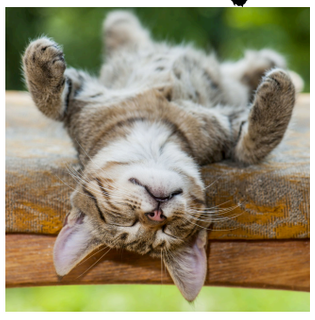
?

Testing Data



Trained Model (f)

Labels



“Cat” (95%)

“Cat” (95%)

“Cat” (85%)

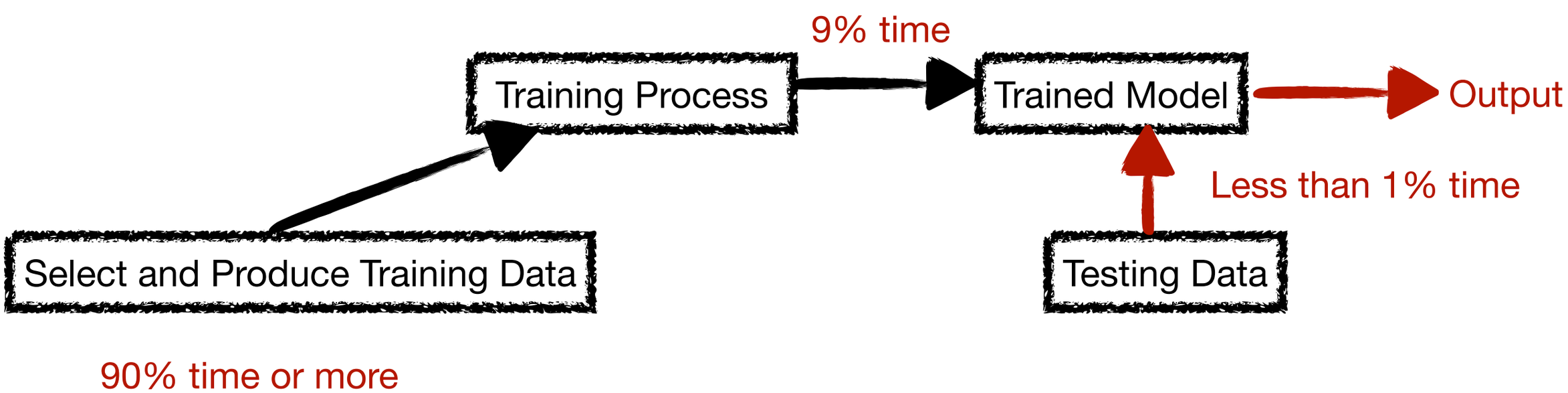
“Unknown” (what’s this guy?)

Training Data

- **Artificial intelligence (AI) is only as good as the data it is trained with**
  - 80% of the time spent on an AI project is wrangling training data, including data labeling
  - Both quality and quantity of training data determine the success of AI

Training Data

- **Artificial intelligence (AI) is only as good as the data it is trained with**
  - 80% of the time spent on an AI project is wrangling training data, including data labeling
  - Both quality and quantity of training data determine the success of AI



Data Labeling

- **Data Labeling**
  - A central part of the data preprocessing workflow for machine learning
  - Defined as the task of detecting and tagging data with labels
  - Give a machine learning model information about what is shown in order to teach the model from these examples
  - Data labeling structures data to make it meaningful
  - After training, able to find “meaning” in new, relevantly similar data.

Simulating Human Learning

Knowledge

Computer Science

Computer Engineering

Earth Science

Meteorology

Labeling

Simulating Human Learning

Knowledge

Computer Science

Computer Engineering

Earth Science

Meteorology

Become familiar with or an expert in an area

Labeling

Inference

Labeling Example (1)

Twitter 1: I’m gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.”,,,

Ham

Twitter 2:,Oh k...i'm watching here:),,,

Ham

Tweet 3: ”SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info”,,,

Spam

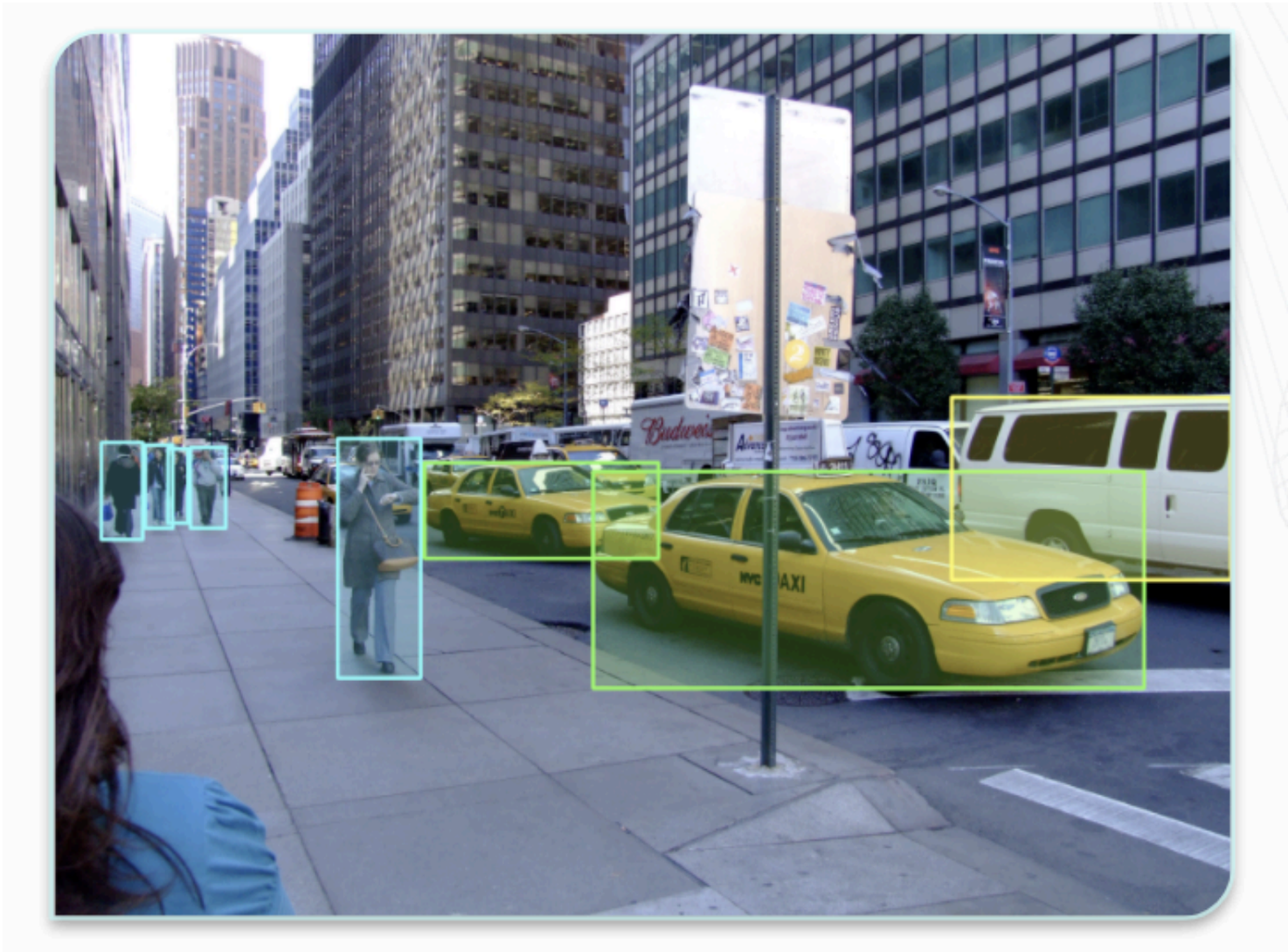
Twitter 4,"URGENT! You have won a 1 week FREE membership in our å£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18”,,,

Spam

Tweet 5,”XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL",,,

Spam

Labeling Example (2)



Source: https://labelbox.com/data-labeling-overview

Labeling Example (3)

N-gram model

You	have	won	free	membership

You	have	won	free	membership

You	have	won	free	membership

You	have	won	free	membership

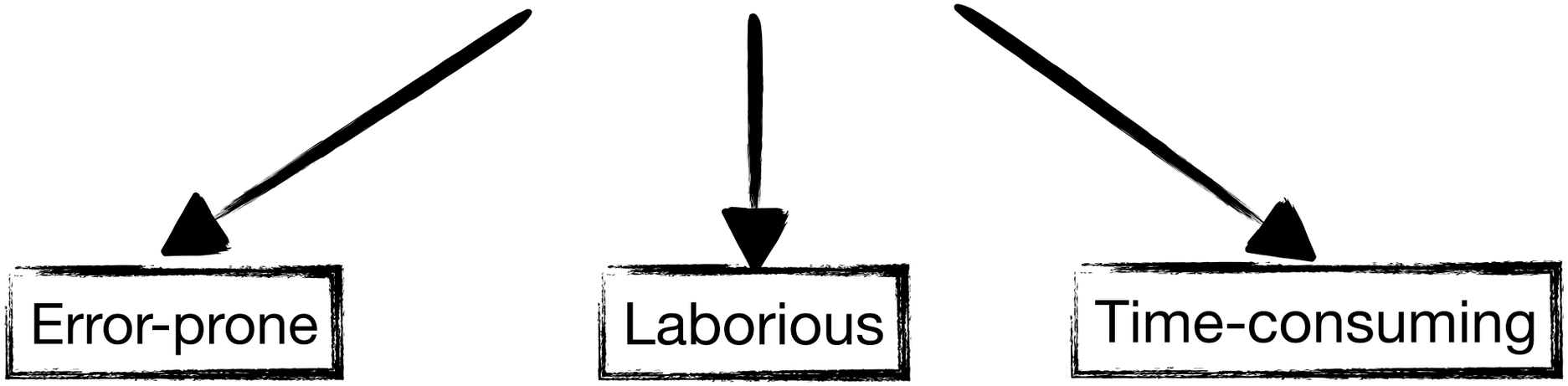
From Previous Coding Practice

```
from sklearn import svm

X = [[0, 1],[1, 2],[2, 1],[2, 3],[1, 3],[2, 2]]

y = ['a', 'a', 'b', 'b', 'a', 'b']

clf = svm.SVC()

clf.fit(X, y)

result1 = clf.predict([[3, 1]])

print(result1)

result2 = clf.predict([[0, 2]])

print(result2)
```

Labeling

```
['b']
['a']
```

So far, it remains a challenging task to label a large reliable dataset!



Labeling Size

How much data do we need to label? It depends on the learning models.

- **Fine-Tuning (FT)**
  - Most common learning approach
  - Updating the weights of a model by training on a supervised dataset
  - large dataset for every task
- **Few-Shot (FS)**
  - Classifying new data when have only a few training samples
  - major reduction in the need for task-specific data
- **One-Shot (1S)**
  - Classify objects from one samples
  - Common in the real world that human learns a task with one demonstration
- **Zero-Shot (0S)**
  - Classify unseen classes without any training examples
  - Using existing labeled data on new tasks

Tweets Labeling

- **Before labeling, we need to know our task**
  - Detecting the spam and non-spam messages
  - So our label will be spam (indicated as 1) or non-spam (indicated as 0)

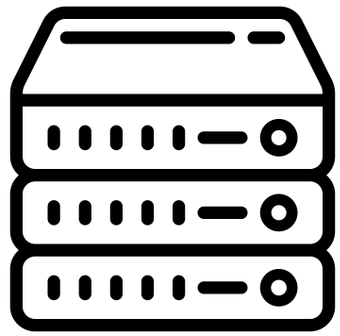
- **A diversified method**
  - Checking suspended account
  - Clustering-based method
  - Rule-based method
  - Manual checking

Checking Suspended Account

- **Suspended Account**

  Check suspended account from twitter.

Twitter API →



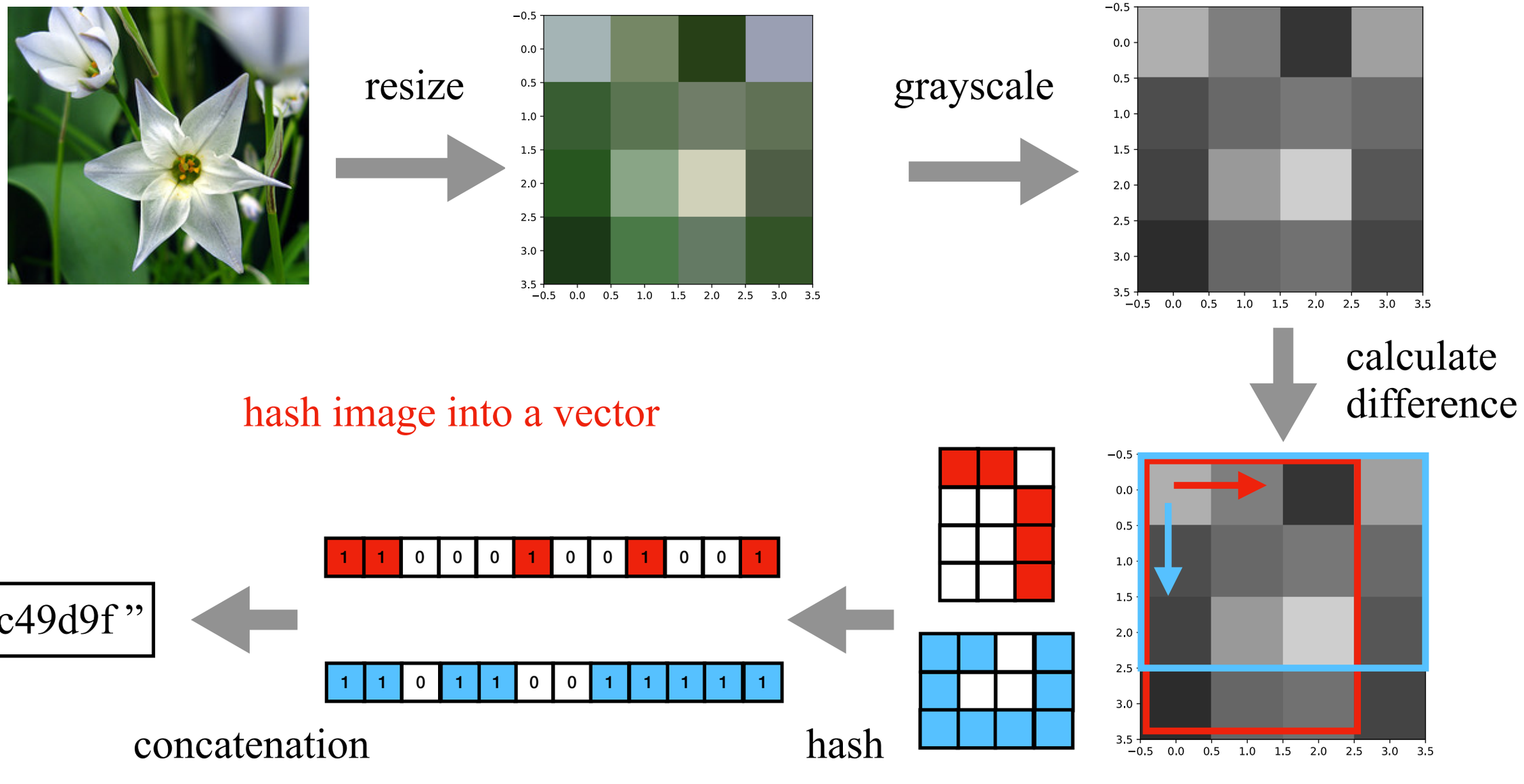
Error Code

50	User not found.
63	User has been suspended.
68	Some actions on this user's Tweet have been disabled by Twitter.
109	The specified user is not found in this list.

Clustering Based Method

- **dHash (1)**

  Cluster near-duplicated images from the social network. However, the images in the social network are not in the same size, and usually very large.

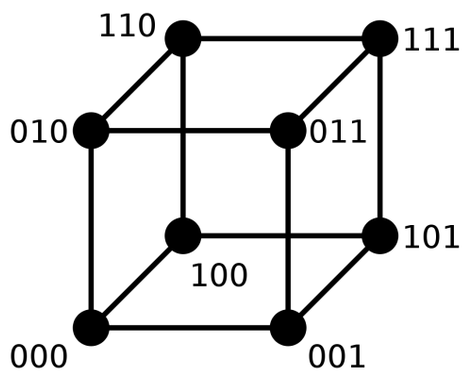


Clustering Based Method

- **dHash (2)**

Hamming Distance

the number of different bits



d(000, 111) = 3

use hamming distance to compare two image hashs

“ c49d9f ”

“ c49d9f ”

d = 0
same cluster

threshold

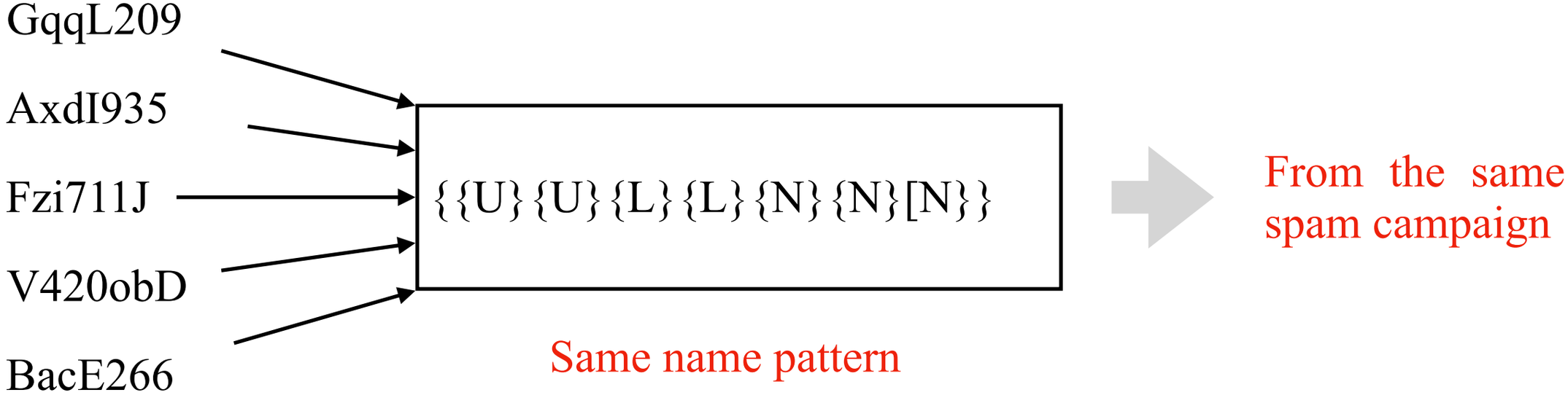
d = 6
not same cluster

“ 88ecd7 ”

Clustering Based Method

- **Automatic Naming Patterns Discovery**

  A spam campaign typically registers its accounts with automatic naming patterns which have relatively limited variability.



Clustering Based Method

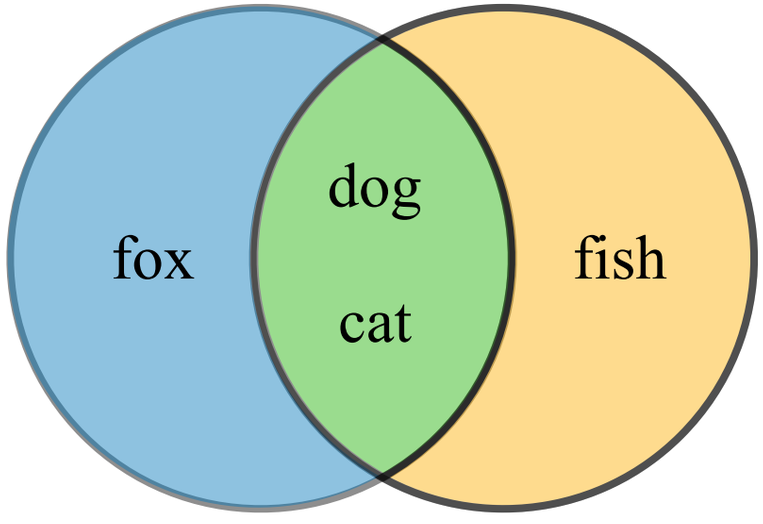
- **minHash (1)**

Cluster near-duplicated content from social networks.

tweet 1: dog, fox, cat

tweet 2: cat, fish, dog

Jaccard similarity



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{2}{4} = 0.5$$

Clustering Based Method

- **minHash (2)**

Cluster near-duplicated content from social network.

Assuming we have N tweets, N-choose-2 comparisons requires:

$$\binom{N}{2} \approx \frac{N^2}{2} \quad \text{comparisons.}$$

A PC can calculate the Jaccard similarity between two sets in 1ms per pair. In twitter, 500 million tweets sent each day.

That means, the total comparison time is

$$\frac{(500 \times 10^6)^2}{2} * \frac{1 \times 10^{-3}}{1 \text{ comparision}} = 7{,}927{,}447 \; years$$

Is there a better solution ?

Clustering Based Method

- **minHash (3)**

Assume we have 3 tweets

t1: dog, fox, cat t2: cat, fish, dog t3: dog, cat, fox

hash function h1
(dog: 1, cat: 3, fish: 5, fox: 4)

hash function h2
(dog: 6, cat: 4, fish: 1, fox: 3)

	t1	t2	t3
h1	1	1	1
h2	3	1	3

minimum hash value

Sim(t1, t2) = 1/2 = 0.5 1 value in common

Sim(s1, s3) = 2/2 = 1 2 value in common

Clustering 600 million tweets

< 1 hour

Can it faster?

Clustering Based Method

- **Single pass clustering**

	t1	t2	t3
h1	1	1	1
h2	3	1	3

C1 = <1, 3> the firth cluster

Sim(C1, t2) = 1/2 = 0.5 assume threshold 0.9

C2 = <1, 1> the second cluster

Sim(C1, t3) = 2/2 = 1

C1 = {t1, t3}

C2 = {t2}

Data labeling

- **Rule-Based Method**

Labeling spam tweets:

1) has malicious URL;
2) includes repetitive information;
3) includes deceptive information;
4) has pertinence purpose;
5) includes many meaningless tweets;
6) has relevant information on free or quick money gain;
7) includes adult content;
8) is an automatic tweet from bots/app with the malicious purpose;
9) is from malicious promoters;
10) is friend infiltrators.
11) includes sensitive or offensive contents.

Labeling ham tweets:

Defining seed accounts:

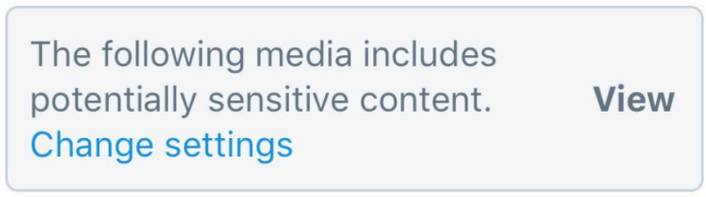
- governments,
- famous companies,
- organizations,
- well-known persons.

Data labeling

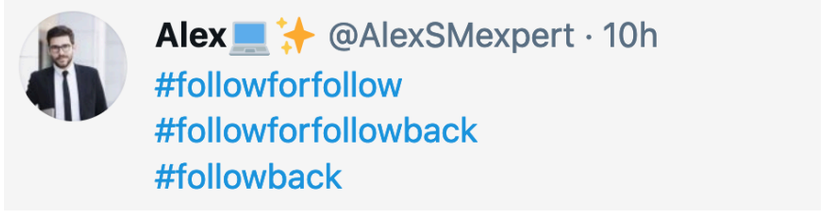
- **Rule-Based Method-Spam Example**

Malicious URL

Sensitive contents



Friend infiltrators



Quick money gain

Rule-Based Method

- **Ham Example**

Governments

Follow

CDC
@CDCgov

CDC's official Twitter source for daily credible health & safety updates from Centers for Disease Control & Prevention. Privacy policy: bit.ly/2MhQ6Hp

Atlanta, GA · cdc.gov · Joined May 2010

Companies

Follow

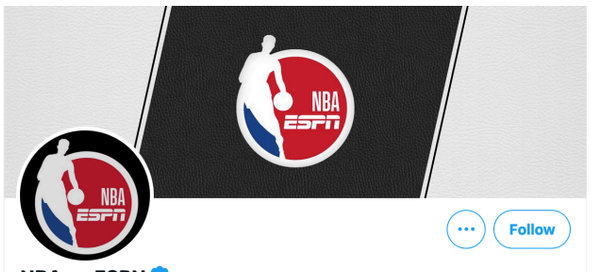
Microsoft
@Microsoft

We're on a mission to empower every person and every organization on the planet to achieve more. Support: @MicrosoftHelps

Redmond, WA · news.microsoft.com · Born April 4, 1975
Joined September 2009

2,538 Following **9.4M** Followers

Organizations



Follow

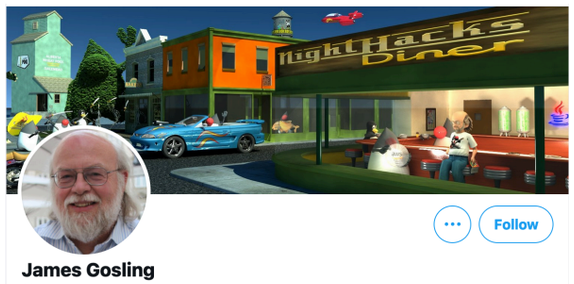
NBA on ESPN
@ESPNNBA

Official Twitter account of the NBA on ESPN, home of the NBA Draft and the NBA Finals. Instagram.com/f/nbaonespn

ESPN · espn.go.com/nba/ · Joined September 2009

695 Following **7M** Followers

People



Follow

James Gosling
@errcraft

Distinguished Engineer at Amazon Web Services

aws.amazon.com · Joined October 2009

66 Following **20.4K** Followers

Data labeling

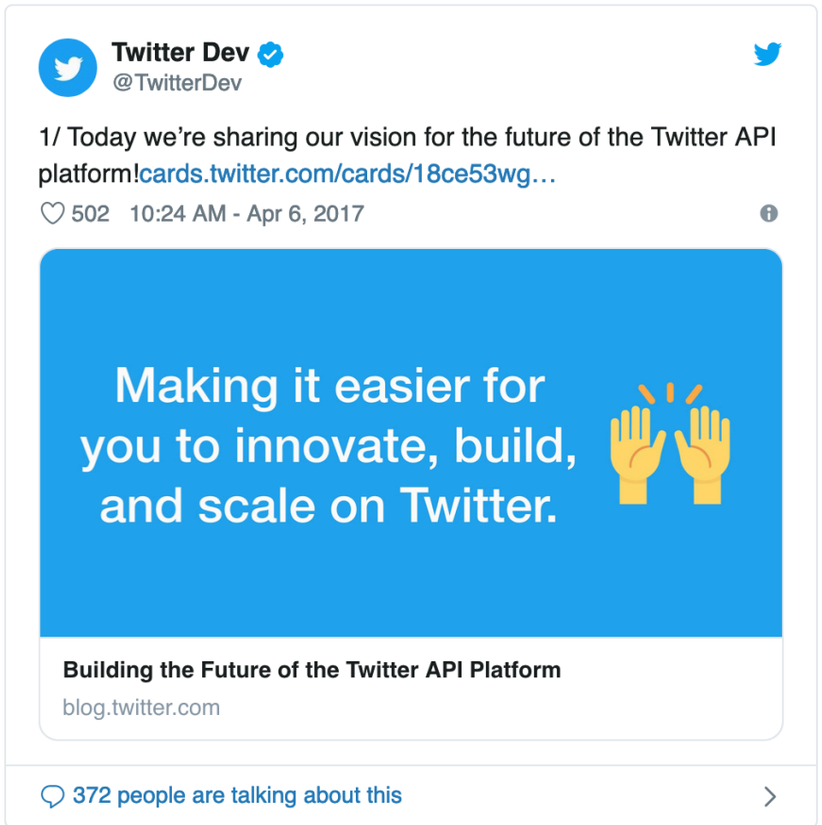
- **Manual checking**

looks like a normal account!

Mimic Normal User

Fraud

Tweet Data Format



Tweet object

Content

Author information

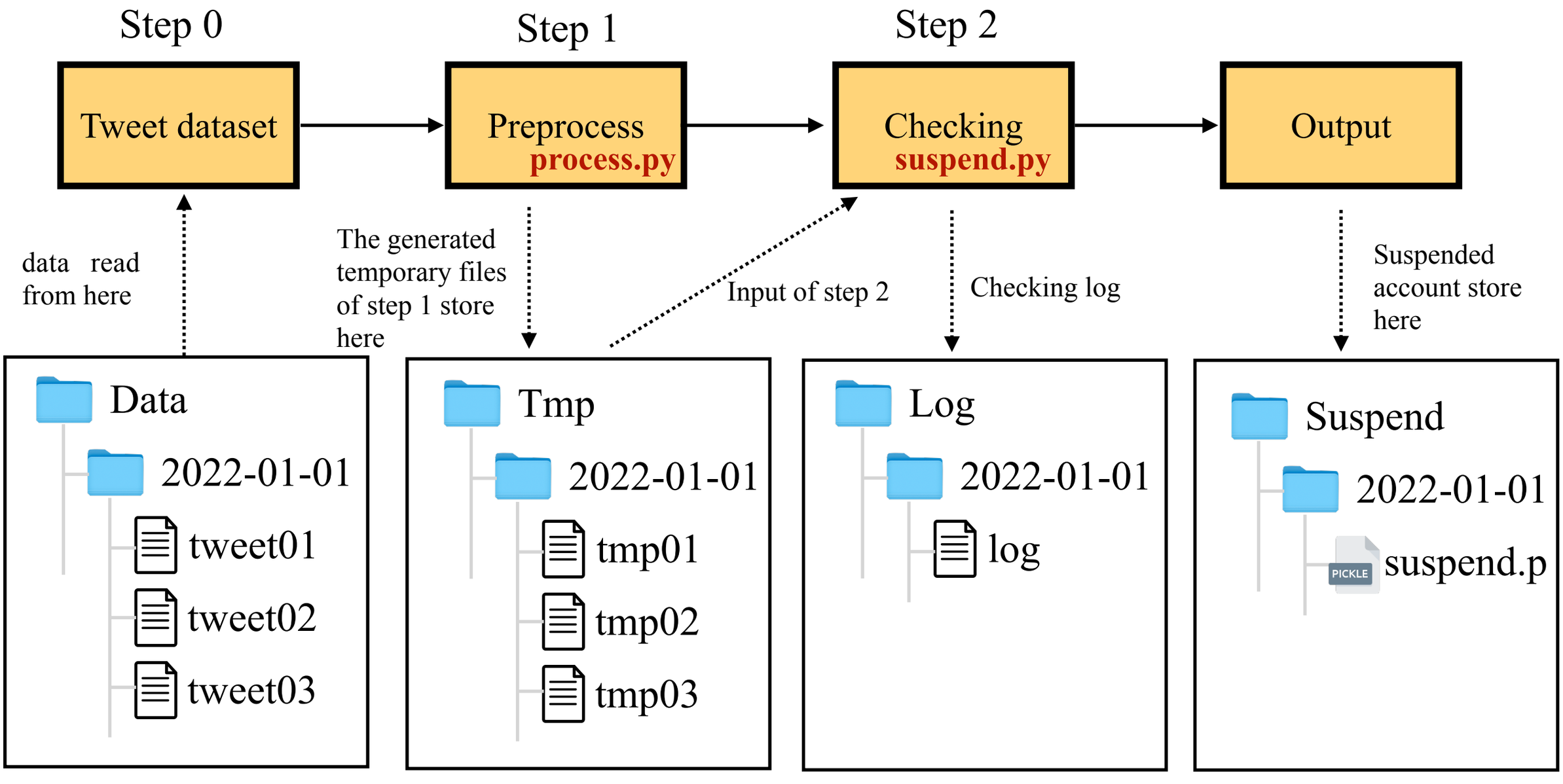
Mentions/Hashtags/URLs


```
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",
  "id_str": "850006245121695744",
  "text": "1\/ Today we\u2019re sharing our vision for the future of
the Twitter API platform!\nhttps:\/\/t.co\/XweGngmxlP",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https:\/\/dev.twitter.com\/",
    "description": "Your official source for Twitter Platform news,
updates & events. Need technical help? Visit https:\/\/
twittercommunity.com\/ \u2328\ufe0f #TapIntoTwitter"
  },
  "place": {
  },
  "entities": {
    "hashtags": [
    ],
    "urls": [
      {
        "url": "https:\/\/t.co\/XweGngmxlP",
        "unwound": {
          "url": "https:\/\/cards.twitter.com\/cards\/18ce53wgo4h\/
3xo1c",
          "title": "Building the Future of the Twitter API Platform"
        }
      }
    ],
    "user_mentions": [
    ]
  }
}
```


Tweet JSON object

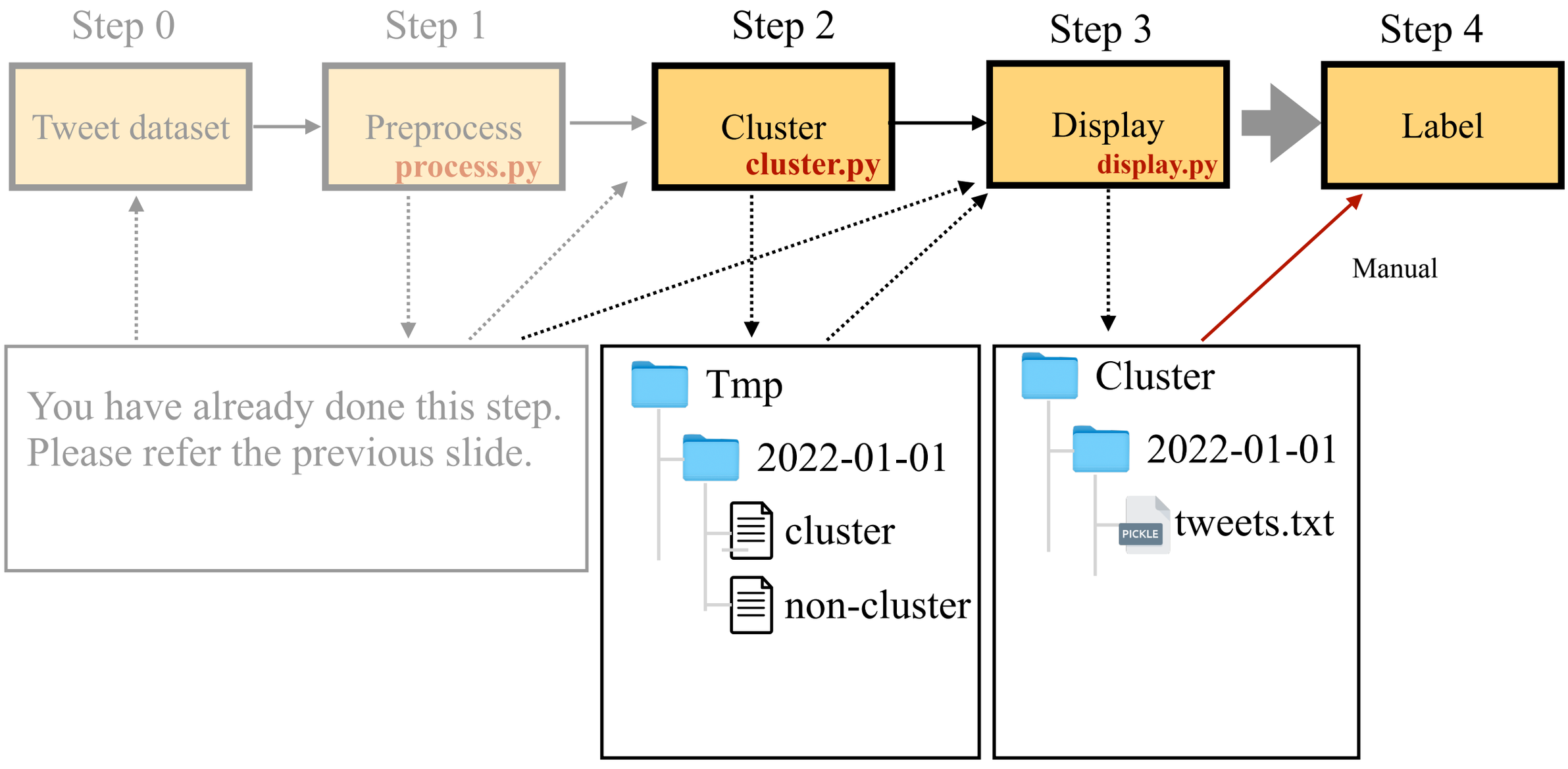
Data labeling

- **Check suspended account**



Data labeling

- **Clustering Tweet**



Data labeling

- **Label Tweet**

Step 3

Display
display.py

Cluster
2022-05-01
PICKLE tweets.txt

Manual Label

Step 4

tweet id	user id	text	label
13***058	26***39	good morning	0
13***059	31***22	free money	1
….			
….			

Assign label for each tweet

0 : non-spam
1 : spam

Coding example

- **Check Suspended Account**

```
start_date = "20210501"
start_date_datetime = datetime.datetime.strptime(start_date, '%Y%m%d')
proc_date = start_date_datetime
duration = 1    # the number of days data

req = ini_req()

def check_status(req, id):
    check_url = 'https://twitter.com/statuses/'+id
    check_res = req.get(check_url)

    if check_res.status_code == 403:
        return True
    else:
        return False

for _ in range(duration):
    # process the data in this date
    proc_date_str = proc_date.strftime("%Y-%m-%d")

    input_data_folder_path = "Tmp/"+proc_date_str+"/"
    output_data_folder_path = "Suspend/"+proc_date_str+"/"+"suspend.p"

    if not os.path.exists(output_data_folder_path):
        os.makedirs(output_data_folder_path)

    suspend_dic = {}

    for filename in os.listdir(input_data_folder_path):
        input_data_path = input_data_folder_path+filename

        with open(input_data_path, 'r', encoding='utf-8', errors='ignore') as file_in:


            for line in file_in:
                tweet_id = line.strip().split("\t")[1]
                is_suspended = check_status(req, tweet_id)
                if is_suspended:
                    suspend_dic[tweet_id] = 1
                else:
                    suspend_dic[tweet_id] = 0

    pickle.dump( suspend_dic, open( output_data_folder_path, "wb" ))
```

Data labeling

- **Clustering Based Method-dHash Code Example**

```python
def calculate_difference_left(grayscale_image):
    pixels = list(grayscale_image.getdata())
    difference = []
    for row in range(resize_height):
        row_start_index = row * resize_width
        for col in range(resize_width - 1):
            left_pixel_index = row_start_index + col
            difference.append(pixels[left_pixel_index] > pixels[left_pixel_index + 1])
    return difference


def calculate_difference_top(grayscale_image):
    pixels = list(grayscale_image.getdata())
    difference = []
    for row in range(resize_height-1):
        row_start_index = row * resize_width
        # print(row_start_index)
        for col in range(resize_width):
            top_pixel_index = row_start_index + col
            difference.append(pixels[top_pixel_index] > pixels[top_pixel_index + resize_width])
    return difference
```

```python
def caculate_hash(difference):
    decimal_value = 0
    hash_string = ""
    for index, value in enumerate(difference):
        if value:
            decimal_value += value * (2 ** (index % 8))
        if index % 8 == 7:
            hash_string += str(hex(decimal_value)[2:].rjust(2, "0"))
            decimal_value = 0
    return hash_string

def hamming_distance(hash1,hash2):
    num = 0
    for index in range(len(hash1)):
        if hash1[index] != hash2[index]:
            num += 1
    return num
```

```python
for f1 in files:
    f1_name = "Image/"+f1
    image1 = Image.open(f1_name)
    smaller_image1 = image1.resize((resize_width, resize_height), Image.ANTIALIAS)
    grayscale_image1 = smaller_image1.convert("L")
    d1_left = calculate_difference_left(grayscale_image1)
    d1_top = calculate_difference_top(grayscale_image1)
    h1_left = caculate_hash(d1_left)
    h1_top = caculate_hash(d1_top)
    h1 = h1_left + h1_top

    for f2 in files:
        if f1 == f2:
            continue

        f2_name = "Image/"+f2
        image2 = Image.open(f2_name)
        smaller_image2 = image2.resize((resize_width, resize_height), Image.ANTIALIAS)
        grayscale_image2 = smaller_image2.convert("L")
        d2 = calculate_difference_left(grayscale_image2)
        h2 = caculate_hash(d2)

        d2_left = calculate_difference_left(grayscale_image2)
        d2_top = calculate_difference_top(grayscale_image2)
        h2_left = caculate_hash(d2_left)
        h2_top = caculate_hash(d2_top)
        h2 = h2_left + h2_top
        dist = hamming_distance(h1, h2)
```