

41. HackerAI

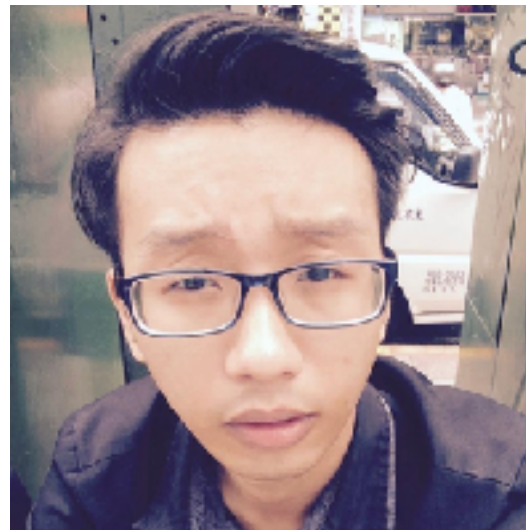
團隊成員



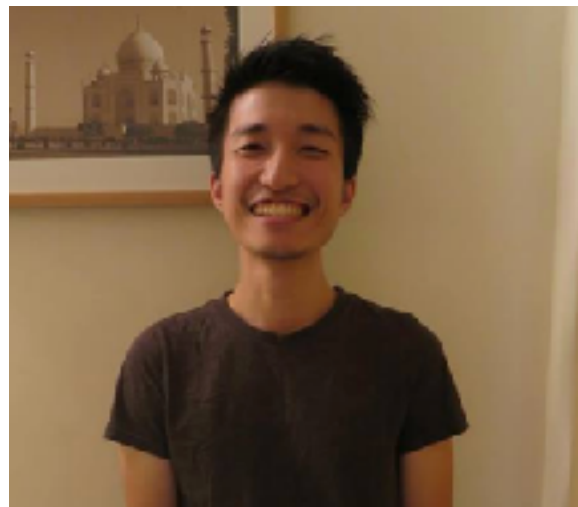
李昀叡
台大電機所
機器學習



羅文斌
台大物理所
數據探勘



顏楷霖
台大生機所
網站全端



汪處剛
清大電機所
深度學習

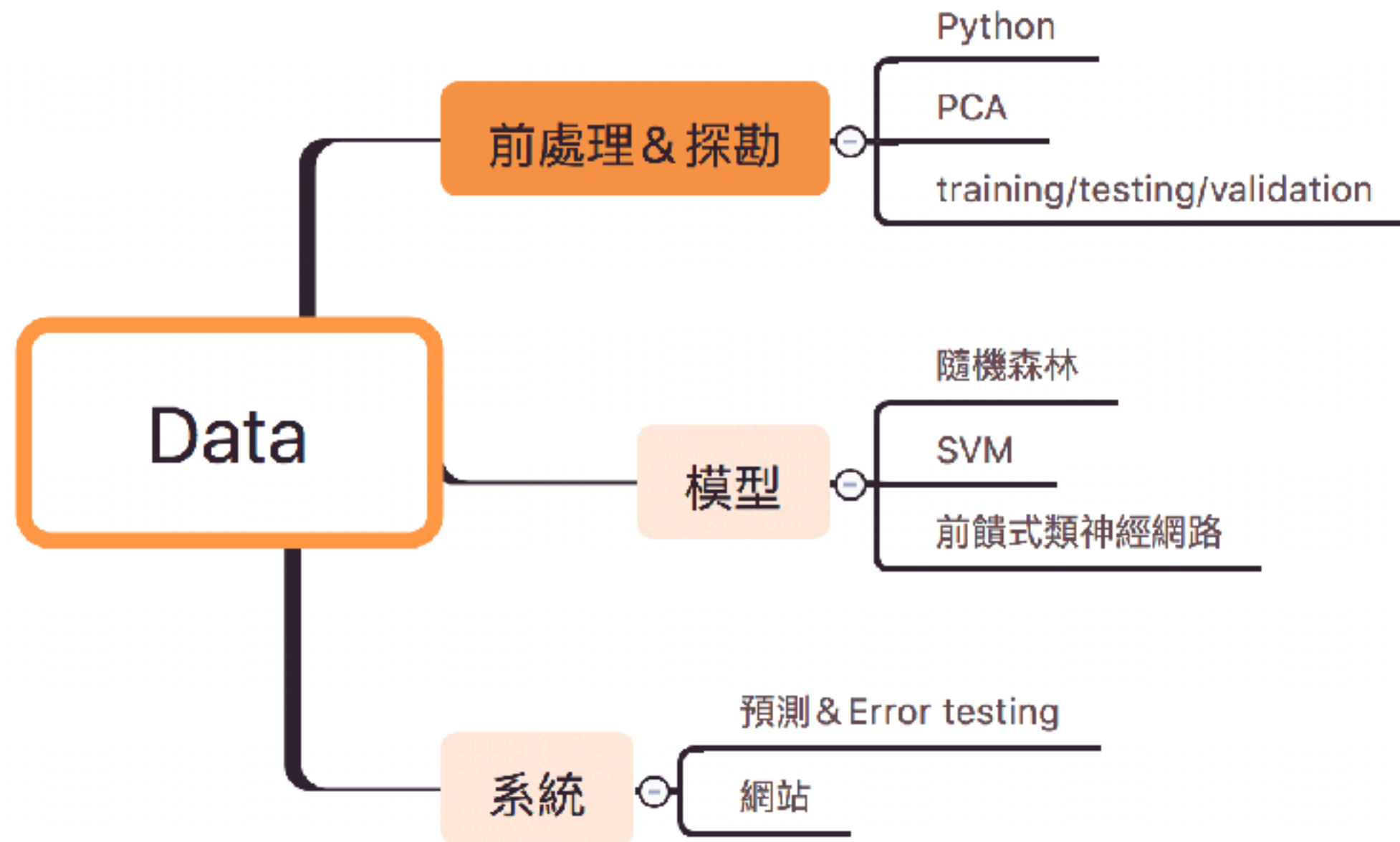


林瑜軒
清大統計所
數據處理

Technology



資料探勘簡介



資料前處理

```
customer_id,efcba383831509f32e1f4ea0c4a09658,purchase_consumption,3696969600,mer
chant_nbr,51821d4f98f7ac1882b32edcd0a33627,credit_card,37fafc5b4f7de0784db066a38
0eb9329,"{"action": {"txn_amt": 1018530.4543357005, "original_currency_code
": "TWD"}, "object": {"consumption_category_desc": "67294eb2089cafae12f8
f919e2520171", "mcc_code_desc": "\\u6df7\\u96dc\\u98df\\u54c1\\u5e97", "me
rchant_category_code": "05499"}, "channel": {"card_type": "674f57e499938
7afb41e10d6bb69d31c", "card_level": "3"}}",3696969600,cc txn
```

```

1 library(r)
2 source("01")
3 source("analysis_04.R")
4
5 #Integration
6 col_names = pd.read_csv('meta/meta_columns.csv').partition_time.columns[0:10].values
7
8 #Integrate all data to one set
9 meta_paths = 'part/partition-time-seconds.csv'
10 sub_paths = [os.path.join(meta_path, v) for v in os.listdir(meta_path) if 'partition' in v]
11
12 full_data = pd.DataFrame()
13 for data_path in sub_paths:
14     data_list = [os.path.join(data_path, y) for y in os.listdir(data_path)]
15     for data in data_list:
16         temp_data = pd.read_csv(data, header=None)
17         full_data = full_data.append(temp_data)
18     full_data = pd.DataFrame(full_data.values.tolist())
19     full_data.to_csv('meta/meta_columns.csv', full_outer=True)
20
21 #data processed
22 col_names = pd.read_csv('meta/meta_columns.csv').columns[0:10].values
23 col_names = [actor_type, 'actor_id', 'action_type', 'action_time', 'object_type', 'object_id',
24             'video_type', 'channel_id', 'key_id', 'category_name', 'category_desc',
25             'cat_code_desc', 'merchant_category_code', 'card_type', 'card_level', 'partition_time', 'time']
26 col_col_name = list(zip(col_names, col_names))

```

```
,customer_id,birth_time,gender,contact_loc,contact_code,register_loc,register_code,start_time,aum,net_profit,credit_card_flag,loan_flag,deposit_f
lag,wealth_flag,partition_time,delta_aum,delta_net_p
317492,00001861a94c52d57aaa71e100f82cff,2832969600,2,1,656911.0,A,656911.0,3624393600,0.0,810511.192393,1,2,2,2,3696969600,0.0,0.0
2619522,00001861a94c52d57aaa71e100f82cff,2832969600,2,1,656911.0,A,656911.0,3624393600,0.0,821983.824343,1,2,2,2,3699475200,0.0,11472.63195
1938341,00001861a94c52d57aaa71e100f82cff,2832969600,2,1,656911.0,A,656911.0,3624393600,0.0,801956.275266,1,2,2,2,3702153600,0.0,-20027.549077
```


ML Features

信用卡消費



MyBank交易



客戶資訊



性別

AUM

居住地

客戶ID

Net_Profit

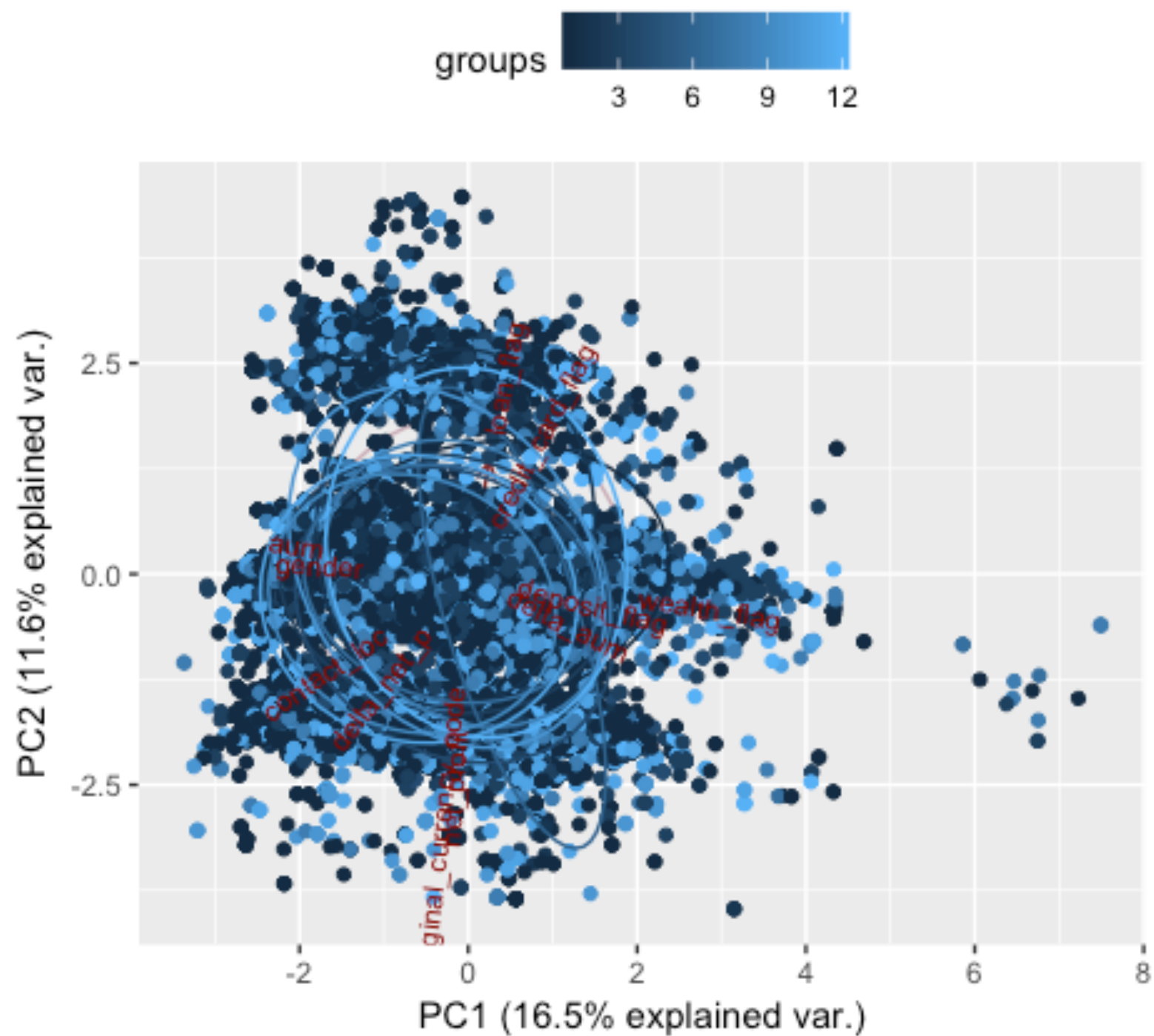
使用服務

Delta
AUM

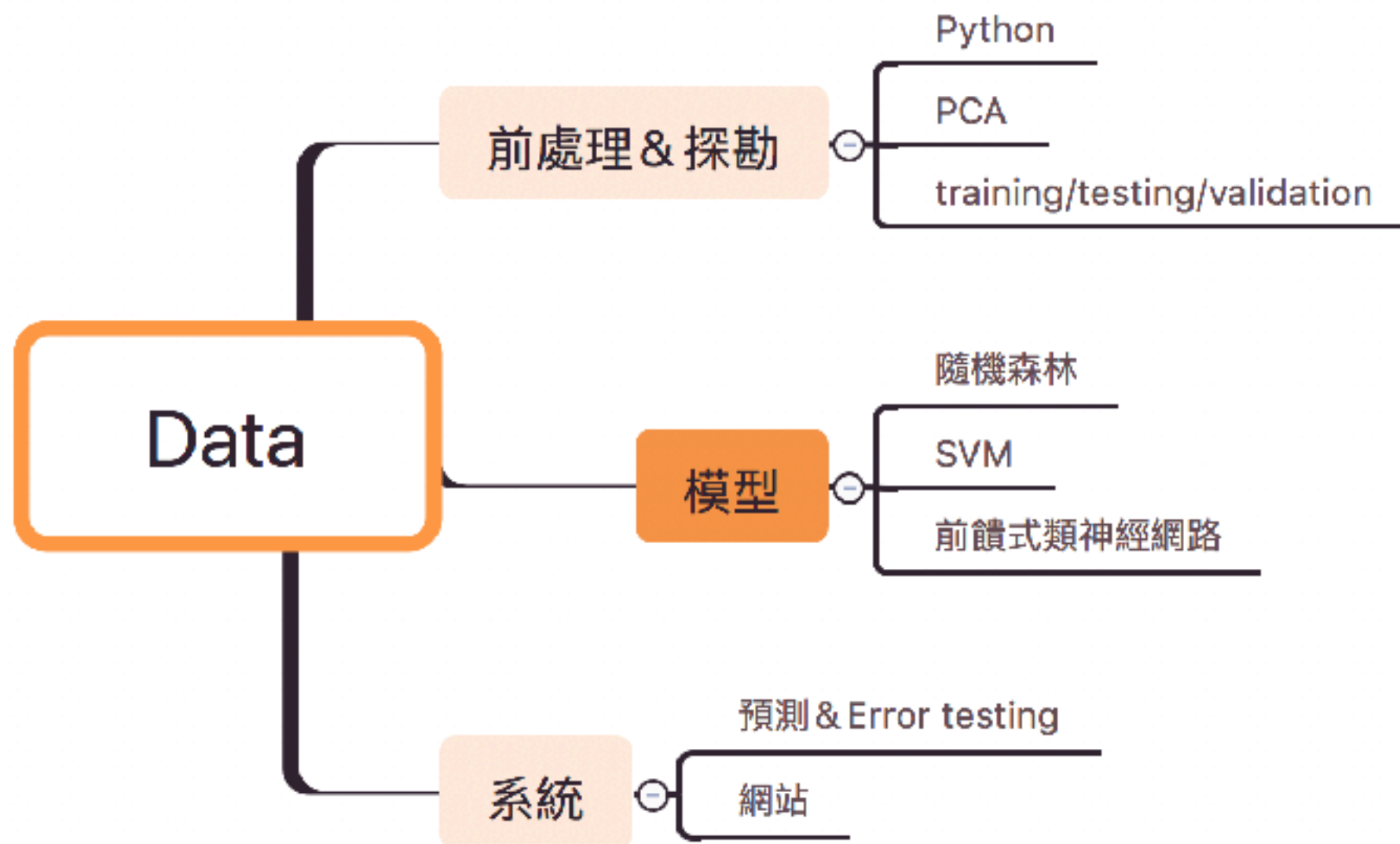
Delta
deNet_Profit

交易幣

資料降維視覺化探索

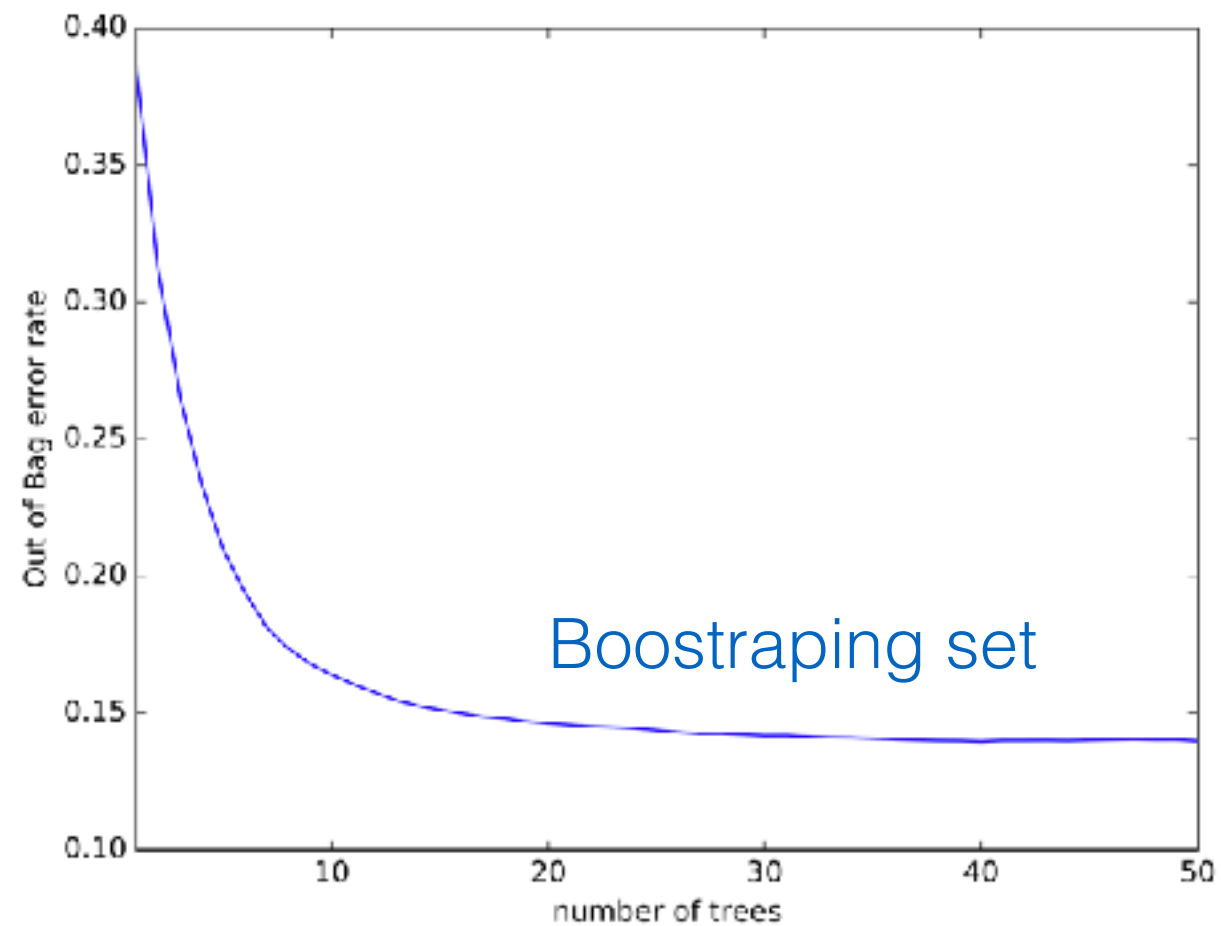


模型介紹

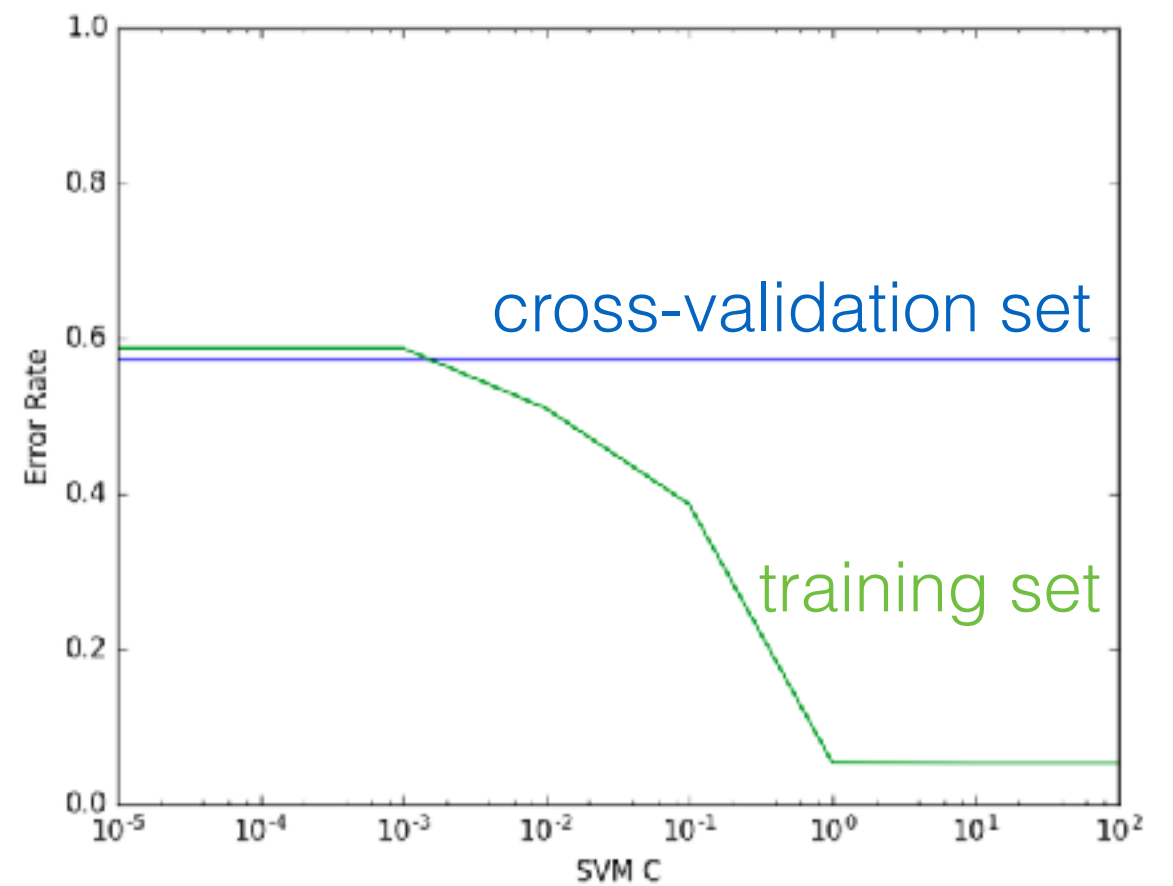


模型結果比較

隨機森林
(正確率~85%)



支撐向量機
(正確率~40%)



模型結果比較

```
def add_layer(inputs, in_size, out_size, activation_function):
    weights = tf.Variable(tf.random_normal([in_size, out_size], mean=0.0, stddev=0.01))
    biases = tf.Variable(tf.zeros([out_size], dtype=tf.float32))
    Wx_plus_b = tf.matmul(inputs, weights) + biases
    if activation_function is None:
        outputs = Wx_plus_b
    else:
        outputs = activation_function(Wx_plus_b)
    return outputs, weights, biases

def compute_accuracy(y, y_pred, sess):
    global losses, acc
    y_pred = sess.run(y_pred)
    correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_pred, 1))
    correct_prediction = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
    result = sess.run([correct_prediction, losses])
    return result

def training_step(sess, inputs, weights, biases, weights_2, biases_2, weights_3, biases_3):
    global losses, acc
    loss = sess.run(losses)
    correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_pred, 1))
    correct_prediction = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
    result = sess.run([correct_prediction, losses])
    return result

def data_loader():
    global data_loader
    data_loader = tf.nn.read_from_tensor_buffer(data_loader)
    return data_loader

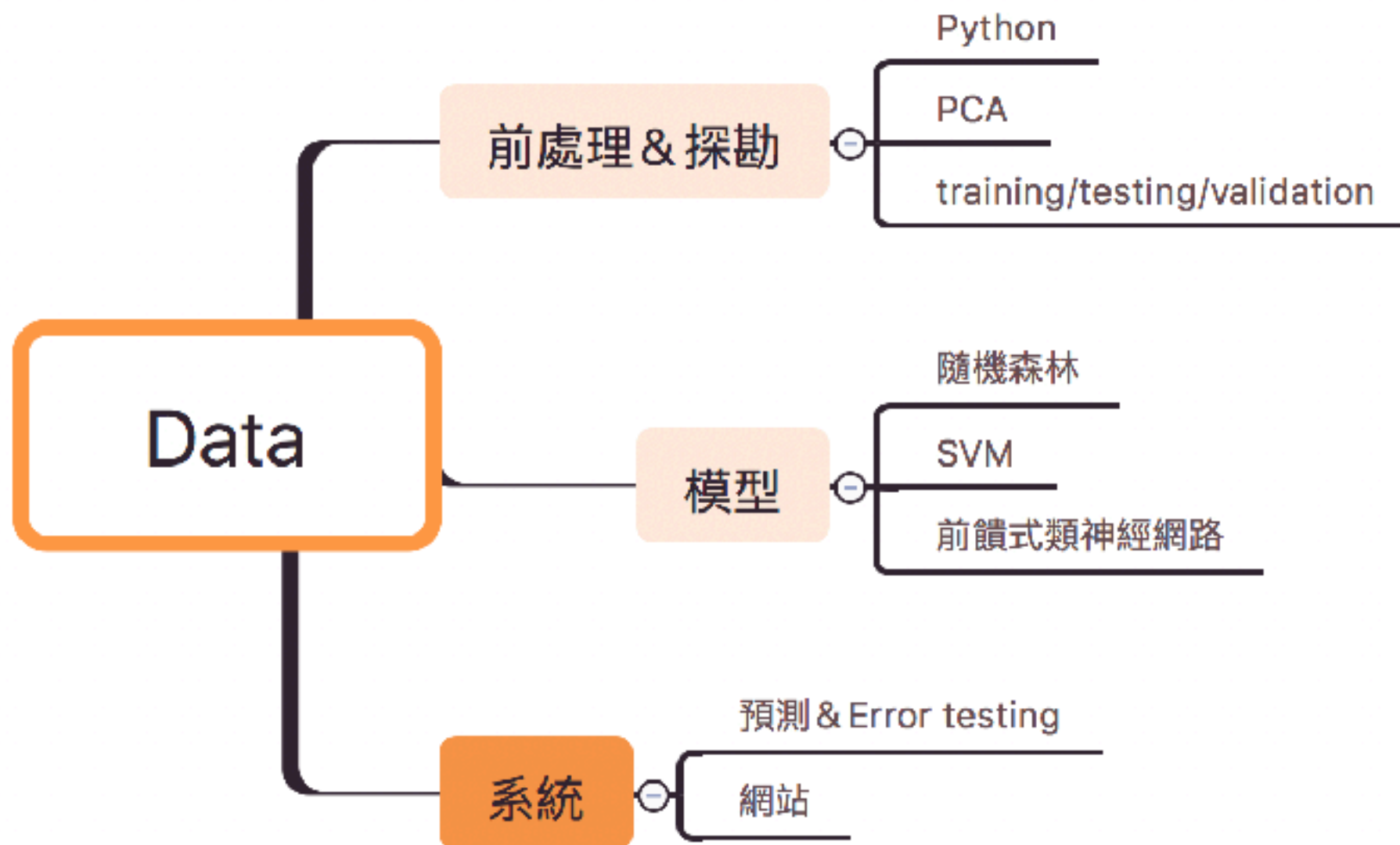
def content_to_data_loader():
    global data_loader
    data_loader = tf.nn.read_from_tensor_buffer(data_loader)
    return data_loader
```

DNN
(正確率~50%)



人類隨機亂猜
(正確率=1/12~8.3%)

系統Demo



網站Demo



HACKERAI	性別	通訊地址	AUM	NET PROFIT
AUM變化量(月)		NET PROFIT變化量(月)		
信用卡	貸款	存款	理財商品往來	
		有的		
		沒有做		
快告訴我結果				

HACKERAI	性別	通訊地址	AUM	NET PROFIT
AUM變化量(月)		NET PROFIT變化量(月)		
信用卡	貸款	存款	理財商品往來	
		有的		
		沒有做		
快告訴我結果				

國泰X台大駭客松推薦信用卡

推薦「3e6282ace882e04800d4b5c458d11269」信用卡給重要的您

OK

2017 FIGHTS FOR

FINTECH

HackerAI 幫你找到數據大金礦

