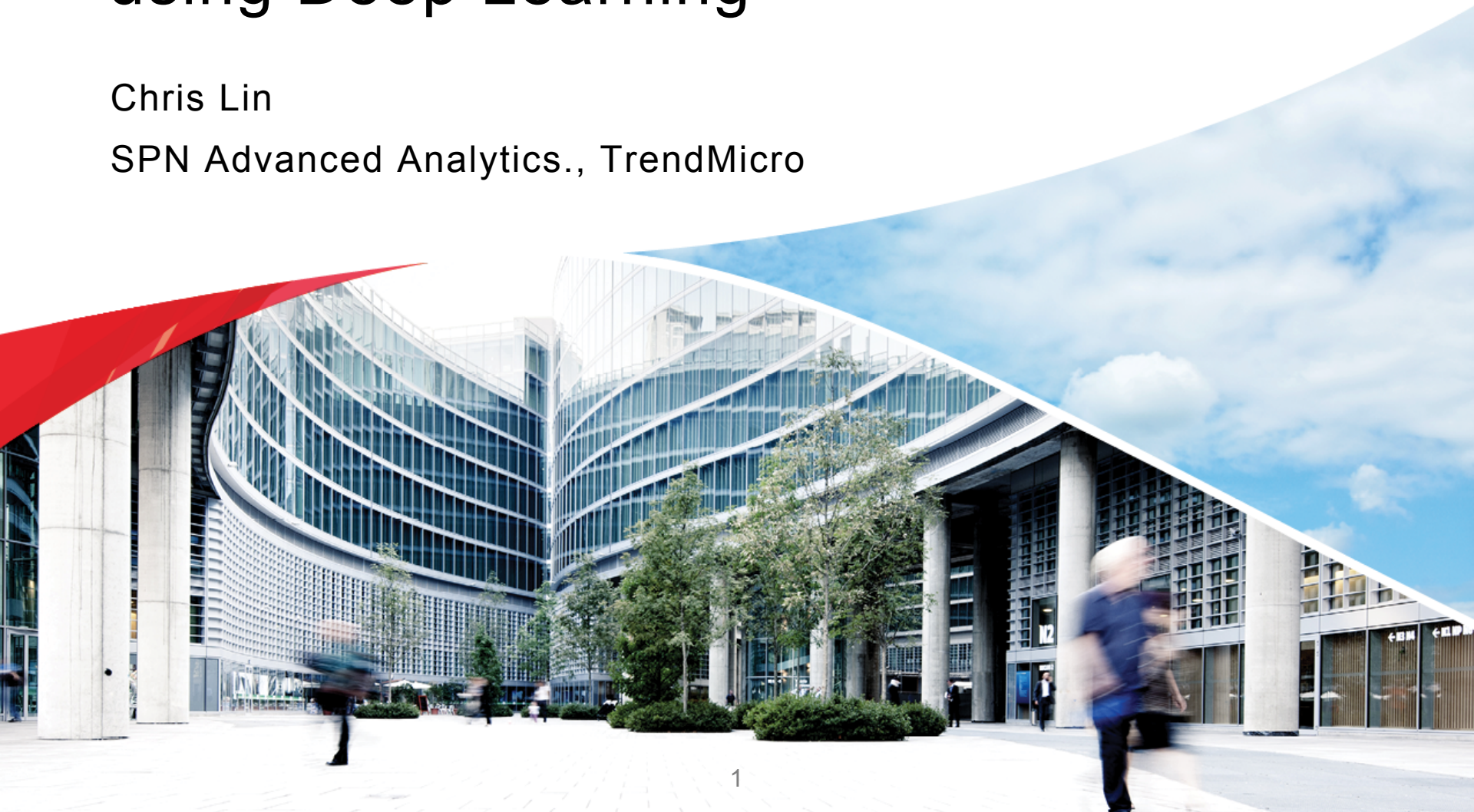
# 104 Job Match Recommendation using Deep Learning

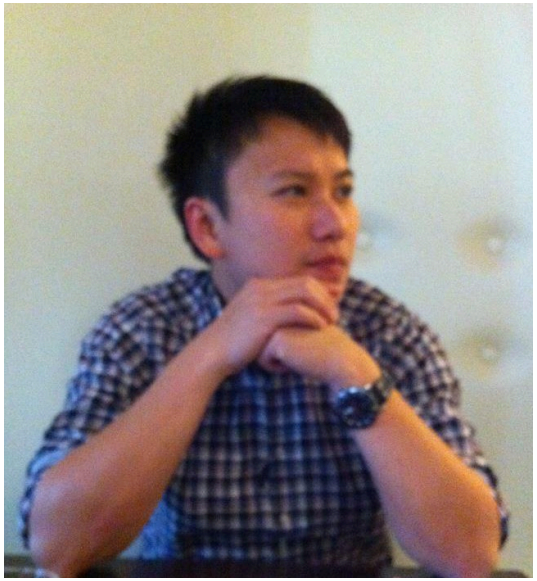
Chris Lin

SPN Advanced Analytics., TrendMicro



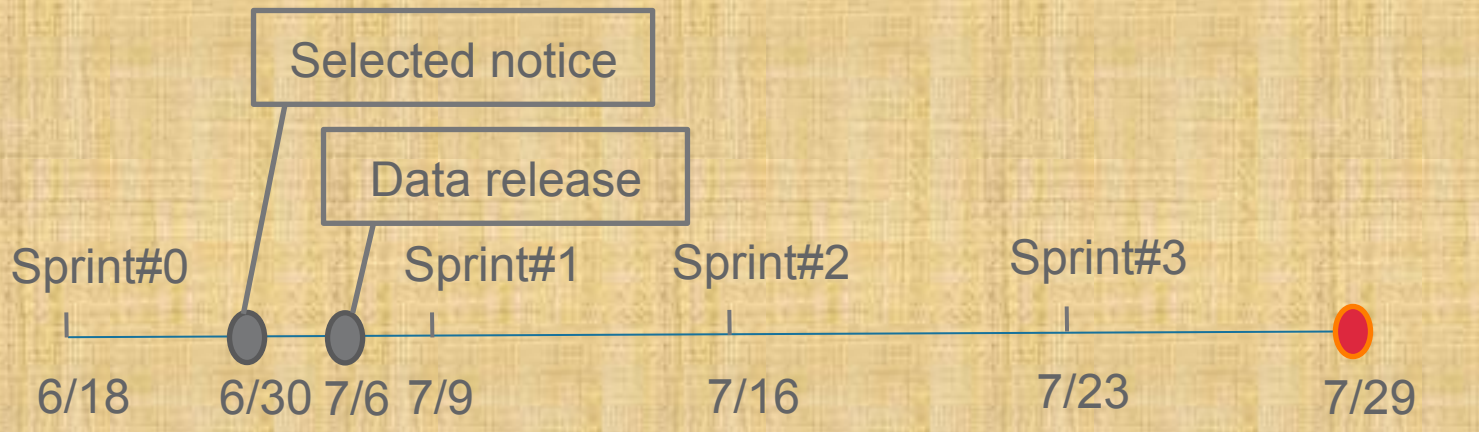
# Lecturer


- Chris Lin
- Sr. Software Engineer @Trend Micro
- Enthusiast in
  - Big data technologies
  - Deep learning
- In charge of
  - Threat identification by graph mining
  - Email writing style identification
  - IoT Security Solution



https://www.linkedin.com/in/tsungfulin/

# Timeline




Prepare Environment
Data Exploration

Data exploration
Feature extraction

Train models

- CNN
- LSTM
- MLP + VAE
- All-in-one

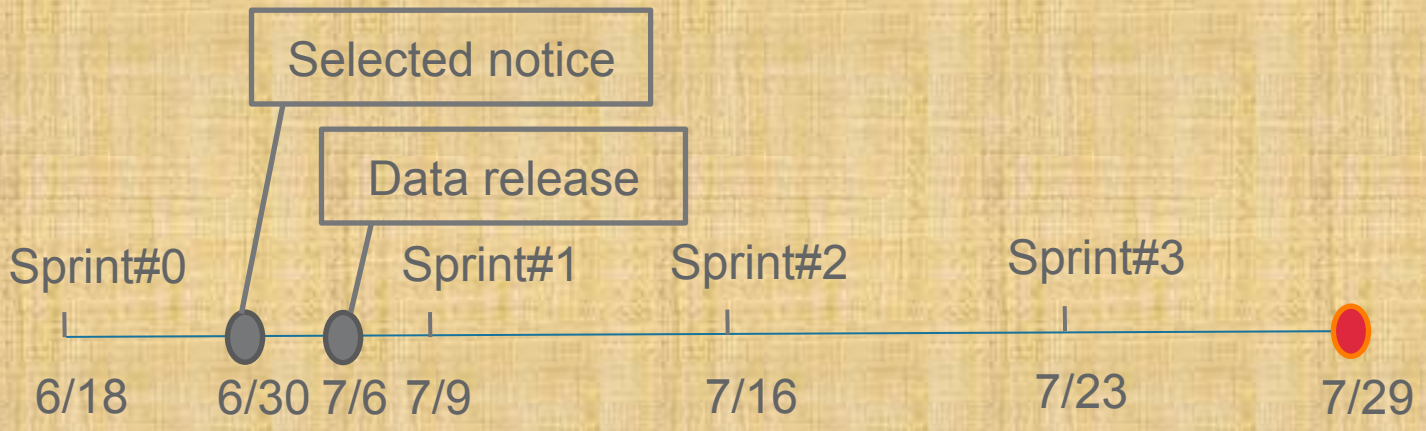
Hyper-parameter and ensemble models

2nd

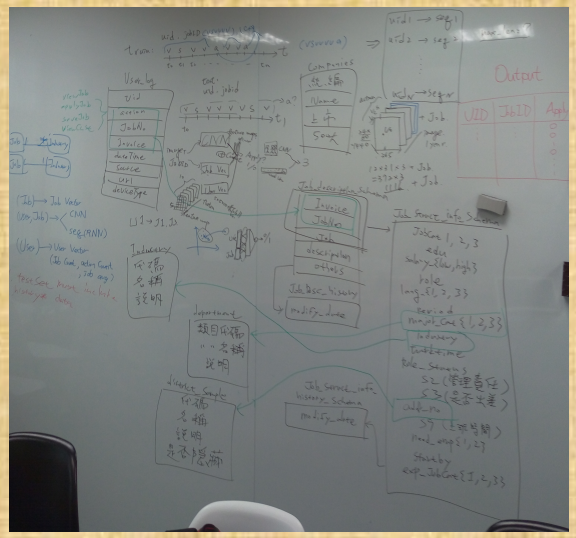
AWS

TREND MICRO

# Timeline <Sprint#0, 1>




Prepare Environment
Data Exploration



Data exploration
Feature extraction

Train models」

- CNN
- LSTM
- MLP + VAE
- All-in-one

Hyper-parameter and ensemble models

2nd

AWS

# 104 Job Match Recommendation

- Task
  - Predict whether user will apply t
- Data
  - User behavior log
    - user_log.csv
  - Job information data
    - company.csv
    - job_structured_info.csv
    - job_description.csv
    - department.csv
    - district.csv
    - industry.csv
    - job_category.csv

| Name | Description |
|---|---|
| invoice | 公司統編 |
| jobno | 職務代碼 |
| job | 職務名稱 |
| jobcat1 | 職務類別1 |
| jobcat2 | 職務類別2 |
| jobcat3 | 職務類別3 |
| edu | 學歷 |
| salary_low | 希望待遇 (LOW) |
| salary_high | 希望待遇 (HIGH) |
| role | 工作型態 |
| language1 | 語言條件1 |
| language2 | 語言條件2 |
| language3 | 語言條件3 |
| period | 工作經驗(年) |
| major_cat | 相關科系類別 |
| major_cat2 | 相關科系類別2 |
| major_cat3 | 相關科系類別3 |
| industry | 公司產業別 |

me|source|url|deviceType
74|2017-03-15 23:59:17|app||1
74|2017-03-17 11:47:35|app||1
-03-16 11:43:24|app||1
74|2017-03-16 11:05:47|app||1
26|2017-03-15 23:58:55|app||1
-03-16 05:23:14|app||1
274||app||1

資料來源: 2017年104資訊科技Hackathon
https://github.com/104corp/2017-104Hackathon-Recommend

TREND MICRO

# Environment

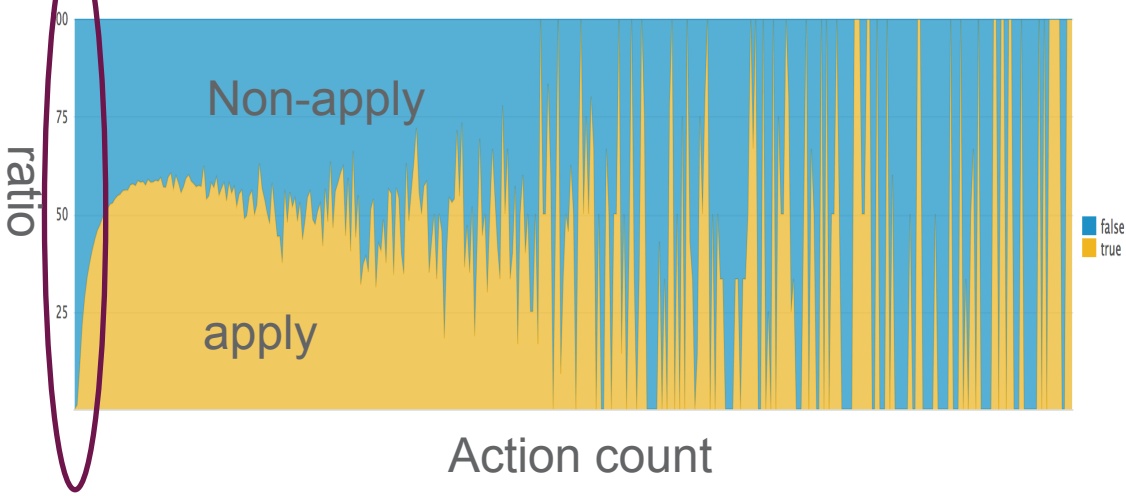
- Data
  - S3 cross account bucket access
- Machine
  - EMR (feature extraction)
  - EC2 (Train model)
- Tool
  - Athena
  - Quicksight
  - Pig
  - Splunk
  - Sklearn, numpy, pandas
  - Keras, Elephas, Hyperas

# Data Exploration

| Action count | Non-apply | apply | apply ratio |
|---|---|---|---|
| 1 | 19171207 | 220855 | 0.011520141 |
| 2 | 5301283 | 861070 | 0.16242525 |
| 3 | 2375958 | 650807 | 0.273708071 |
| 4 | 1388611 | 533568 | 0.38424584 |
| 5 | 876475 | 398085 | 0.454188653 |
| 6 | 608200 | 313405 | 0.515299244 |
| 7 | 432447 | 239267 | 0.553286299 |
| 8 | 325463 | 192660 | 0.591956689 |
| 9 | 247793 | 152698 | 0.616232097 |
| 10 | 195299 | 123798 | 0.633889574 |
| 11 | 154260 | 99845 | 0.647251394 |
| 12 | 125463 | 82394 | 0.656719511 |
| 13 | 102893 | 68097 | 0.661823448 |
| 14 | 85285 | 57265 | 0.671454535 |
| 15 | 71770 | 47996 | 0.668747387 |
| 16 | 60719 | 40767 | 0.671404338 |
| 17 | 51938 | 34836 | 0.670722785 |
| 18 | 44510 | 29645 | 0.666030106 |
| 19 | 38605 | 25486 | 0.660173553 |
| 20 | 33886 | 22170 | 0.654252494 |

action count =1

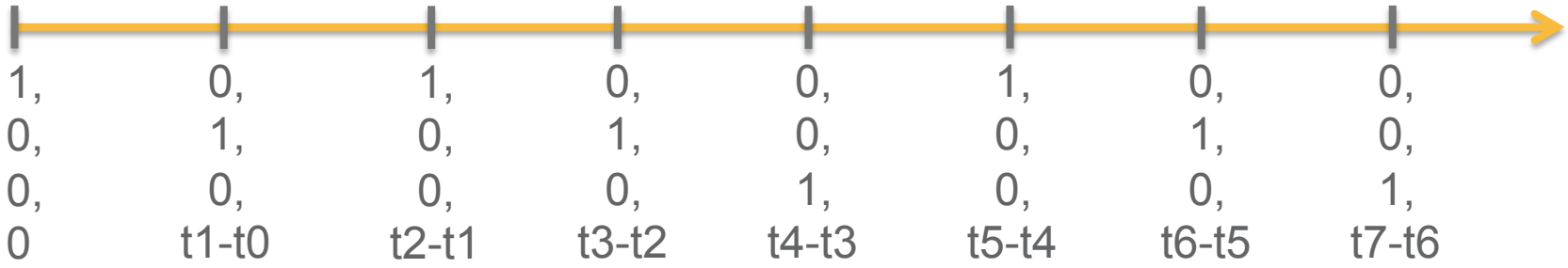
98% Non-apply!!!!!



# Preprocess – User Behavior Log

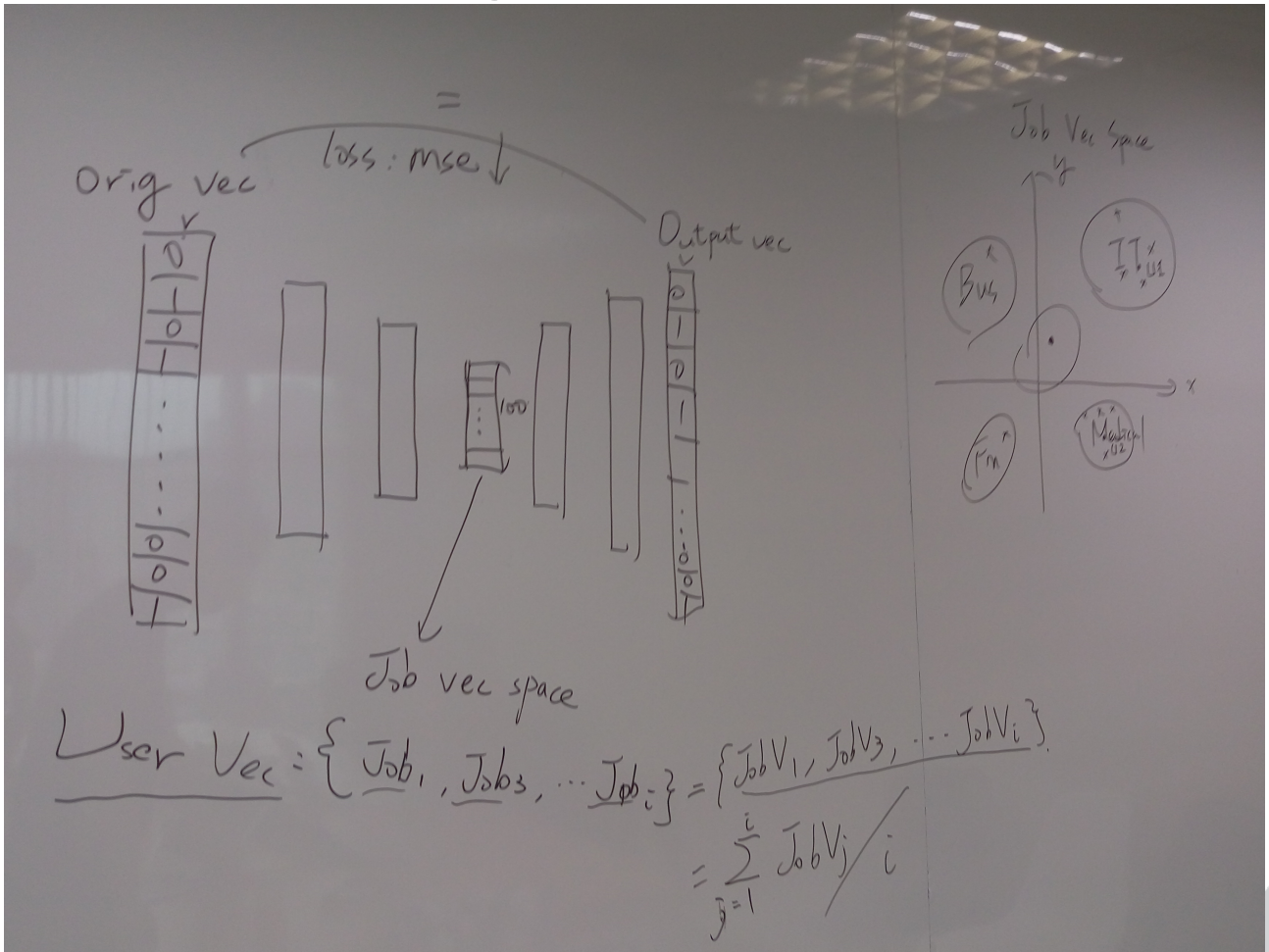
- Group by (uid, JobNo)
- Action Sequence (30*4) → 95% action sequence length < 30
  - 3 actions
    - viewJob (1, 0, 0)
    - saveJob (0, 1, 0)
    - viewCust (0, 0, 1)
  - Time interval

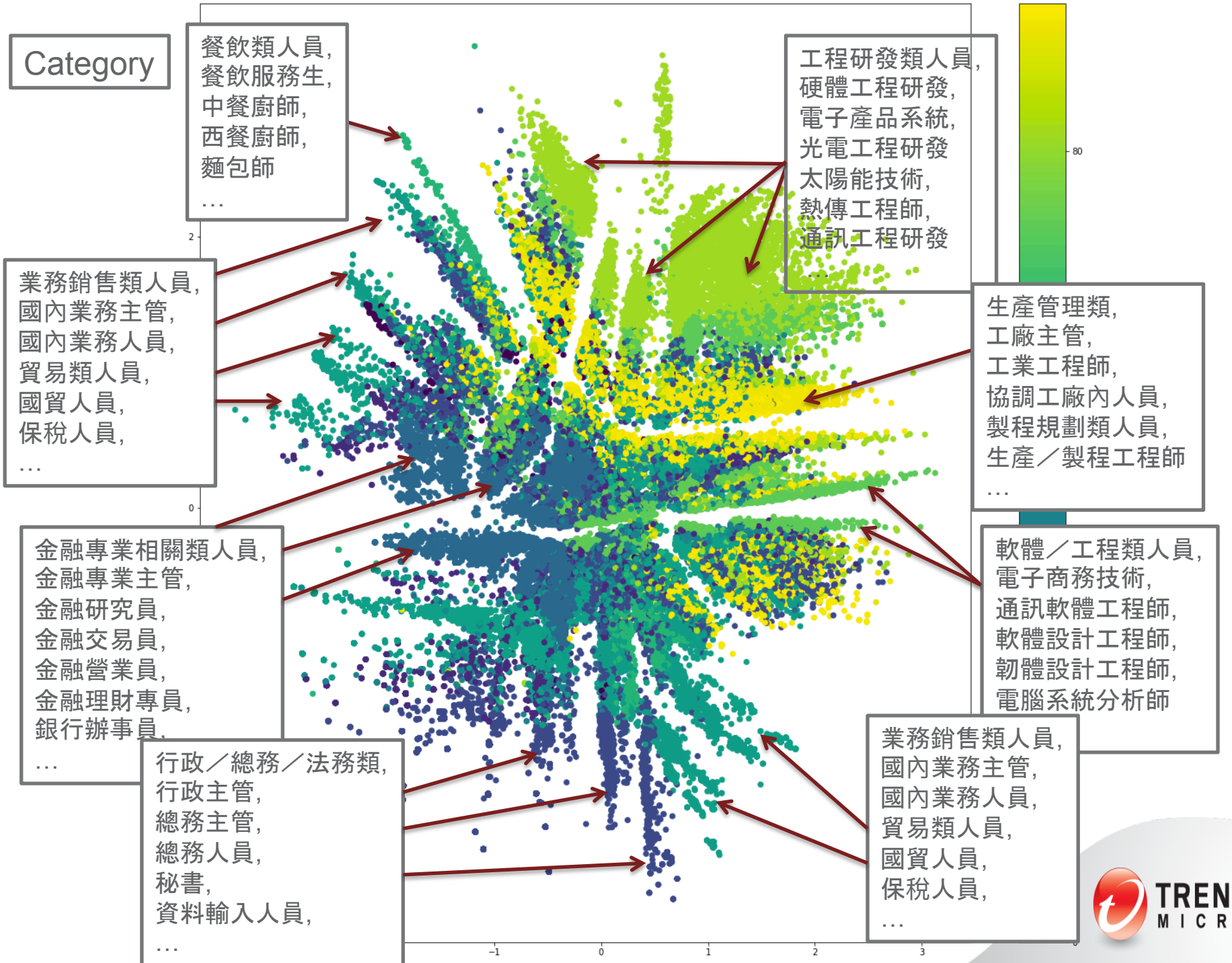
Ex: viewJob, saveJob, viewJob, saveJob, viewCust, viewJob, saveJob, viewCust, …
t0 < t1 < t2 < t3 < t4 < t5 < t6 < t7 …

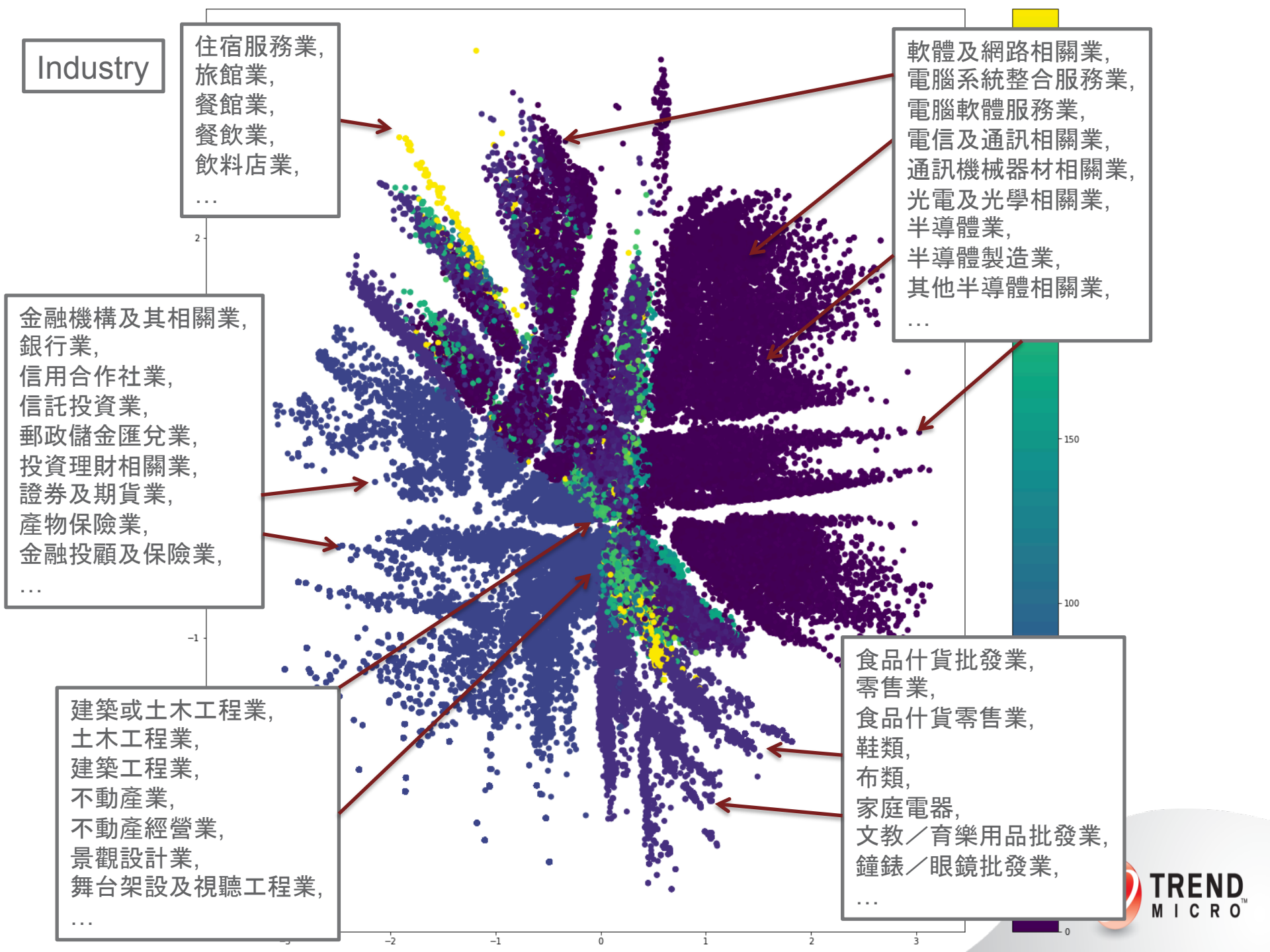


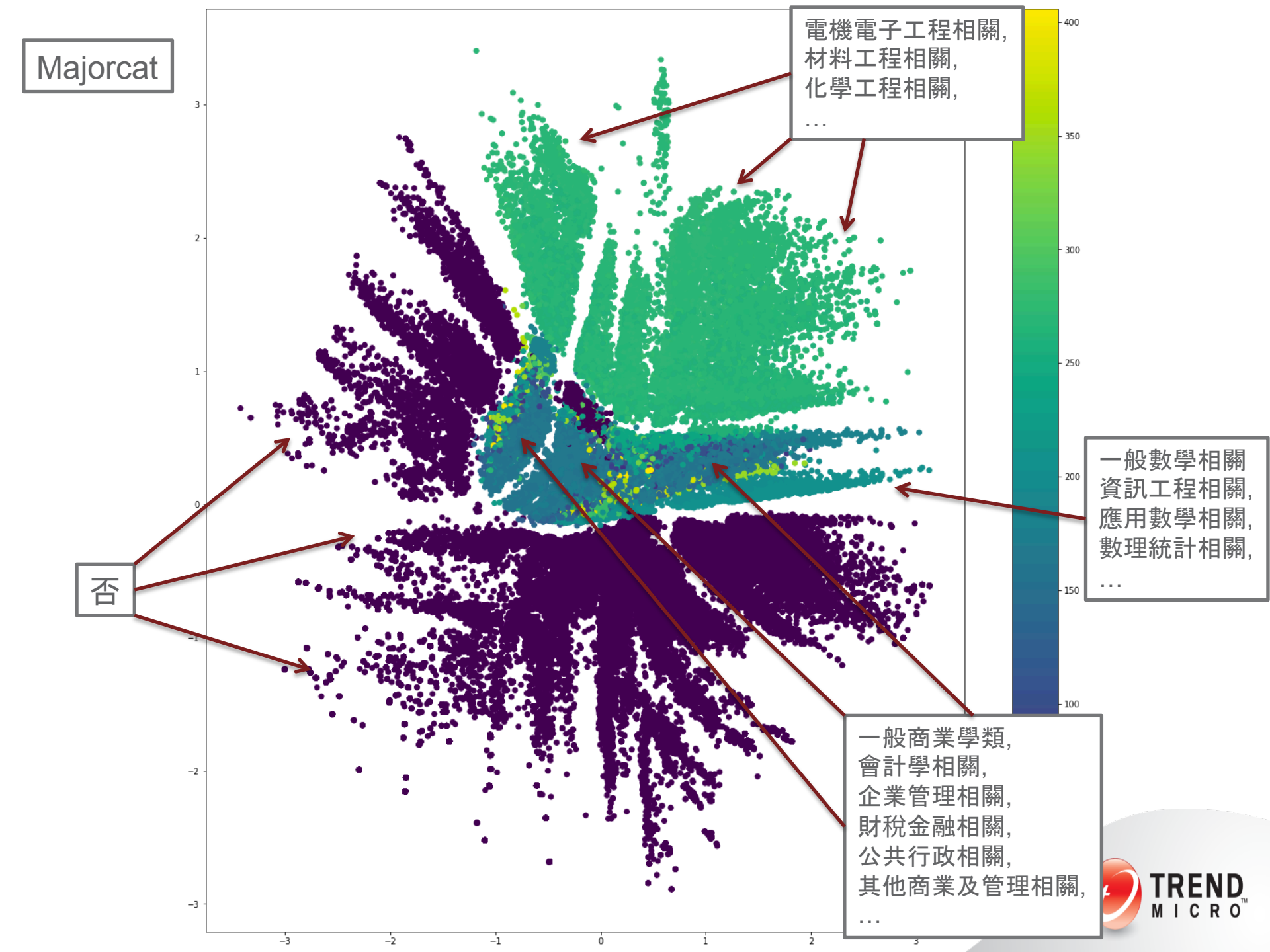
# Preprocess – Job information

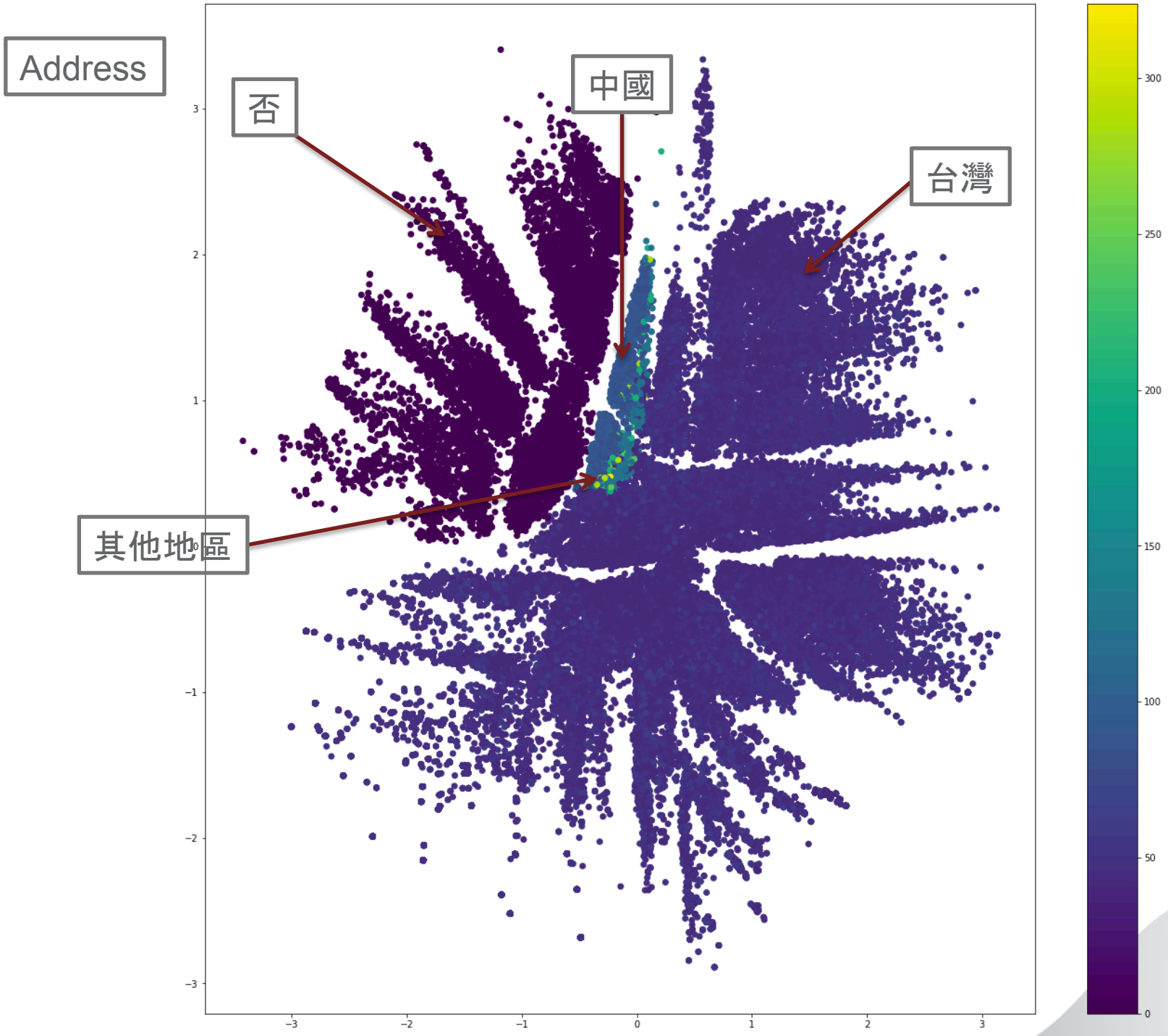
- Get the Job categorical data
  - Jobcat, edu, role, period, language, … etc
- Reduce dimension to get Job hidden space



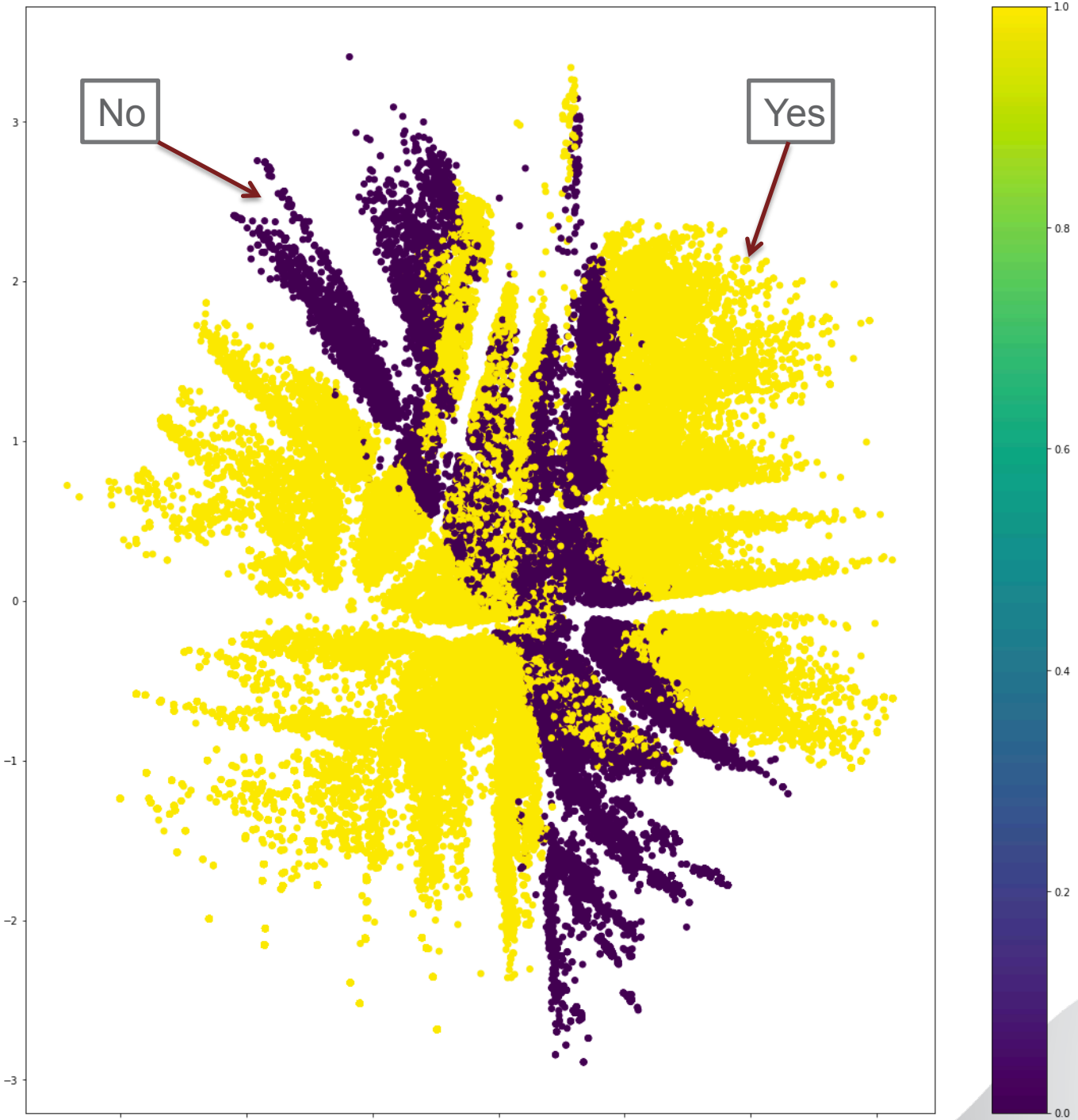

Category
餐飲類人員,
餐飲服務生,
中餐廚師,
西餐廚師,
麵包師
…
工程研發類人員,
硬體工程研發,
電子產品系統,
光電工程研發
太陽能技術,
熱傳工程師,
通訊工程研發
…
業務銷售類人員,
國內業務主管,
國內業務人員,
貿易類人員,
國貿人員,
保稅人員,
…
生產管理類,
工廠主管,
工業工程師,
協調工廠內人員,
製程規劃類人員,
生產／製程工程師
…
金融專業相關類人員,
金融專業主管,
金融研究員,
金融交易員,
金融營業員,
金融理財專員,
銀行辦事員,
…
軟體／工程類人員,
電子商務技術,
通訊軟體工程師,
軟體設計工程師,
韌體設計工程師,
電腦系統分析師
行政／總務／法務類,
行政主管,
總務主管,
總務人員,
秘書,
資料輸入人員,
…
業務銷售類人員,
國內業務主管,
國內業務人員,
貿易類人員,
國貿人員,
保稅人員,
…
80
2
0
−1
0
1
2
3


Industry
住宿服務業,
旅館業,
餐館業,
餐飲業,
飲料店業,
…
軟體及網路相關業,
電腦系統整合服務業,
電腦軟體服務業,
電信及通訊相關業,
通訊機械器材相關業,
光電及光學相關業,
半導體業,
半導體製造業,
其他半導體相關業,
…
金融機構及其相關業,
銀行業,
信用合作社業,
信託投資業,
郵政儲金匯兌業,
投資理財相關業,
證券及期貨業,
產物保險業,
金融投顧及保險業,
…
建築或土木工程業,
土木工程業,
建築工程業,
不動產業,
不動產經營業,
景觀設計業,
舞台架設及視聽工程業,
…
食品什貨批發業,
零售業,
食品什貨零售業,
鞋類,
布類,
家庭電器,
文教／育樂用品批發業,
鐘錶／眼鏡批發業,
…
2
−1
−3
−2
−1
0
1
2
3
150
100
0


Majorcat
電機電子工程相關,
材料工程相關,
化學工程相關,
...
一般數學相關
資訊工程相關,
應用數學相關,
數理統計相關,
...
否
一般商業學類,
會計學相關,
企業管理相關,
財稅金融相關,
公共行政相關,
其他商業及管理相關,
...
TREND MICRO

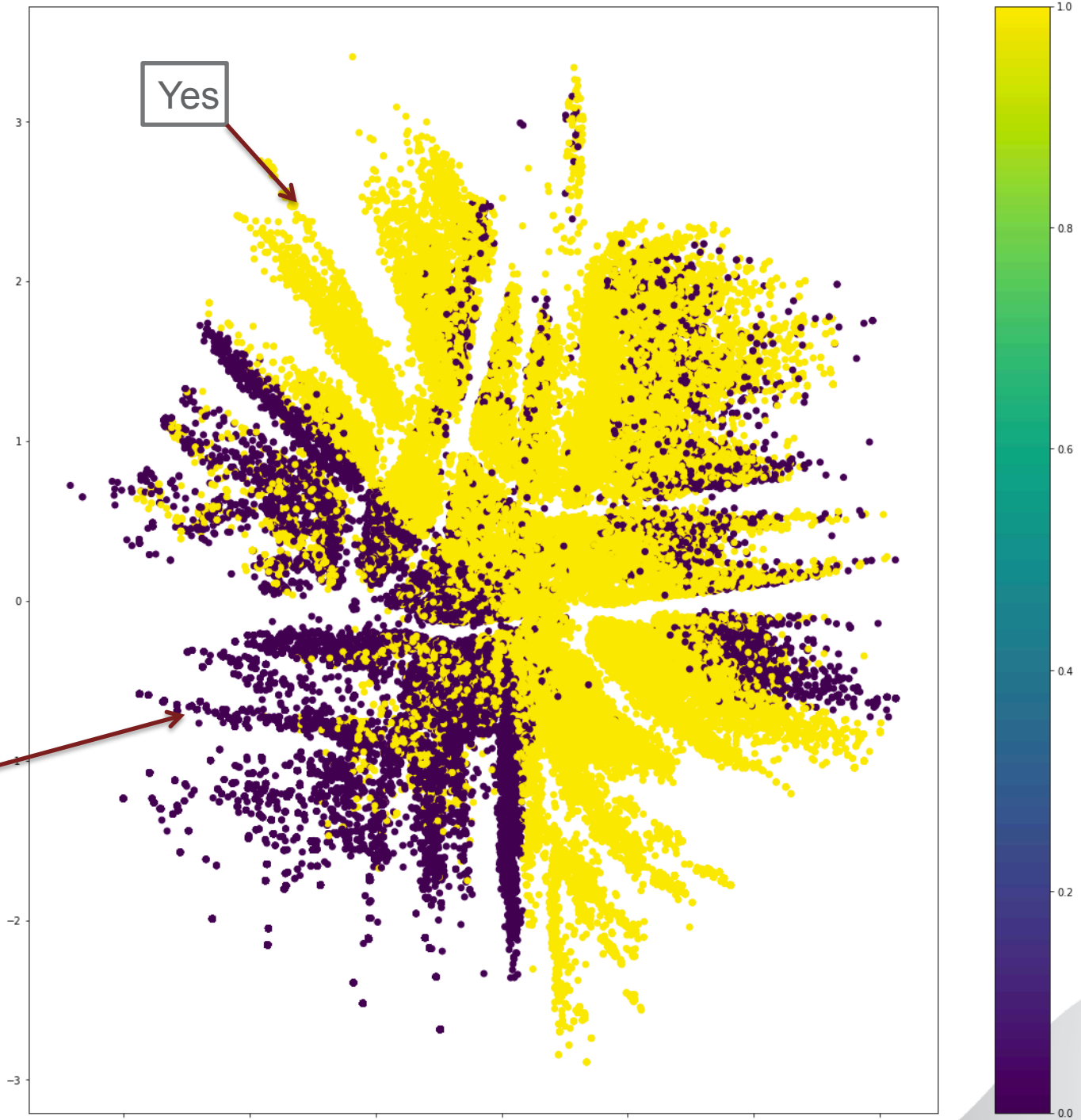

Address
否
中國
台灣
其他地區
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
300
250
200
150
100
50
0

Top500



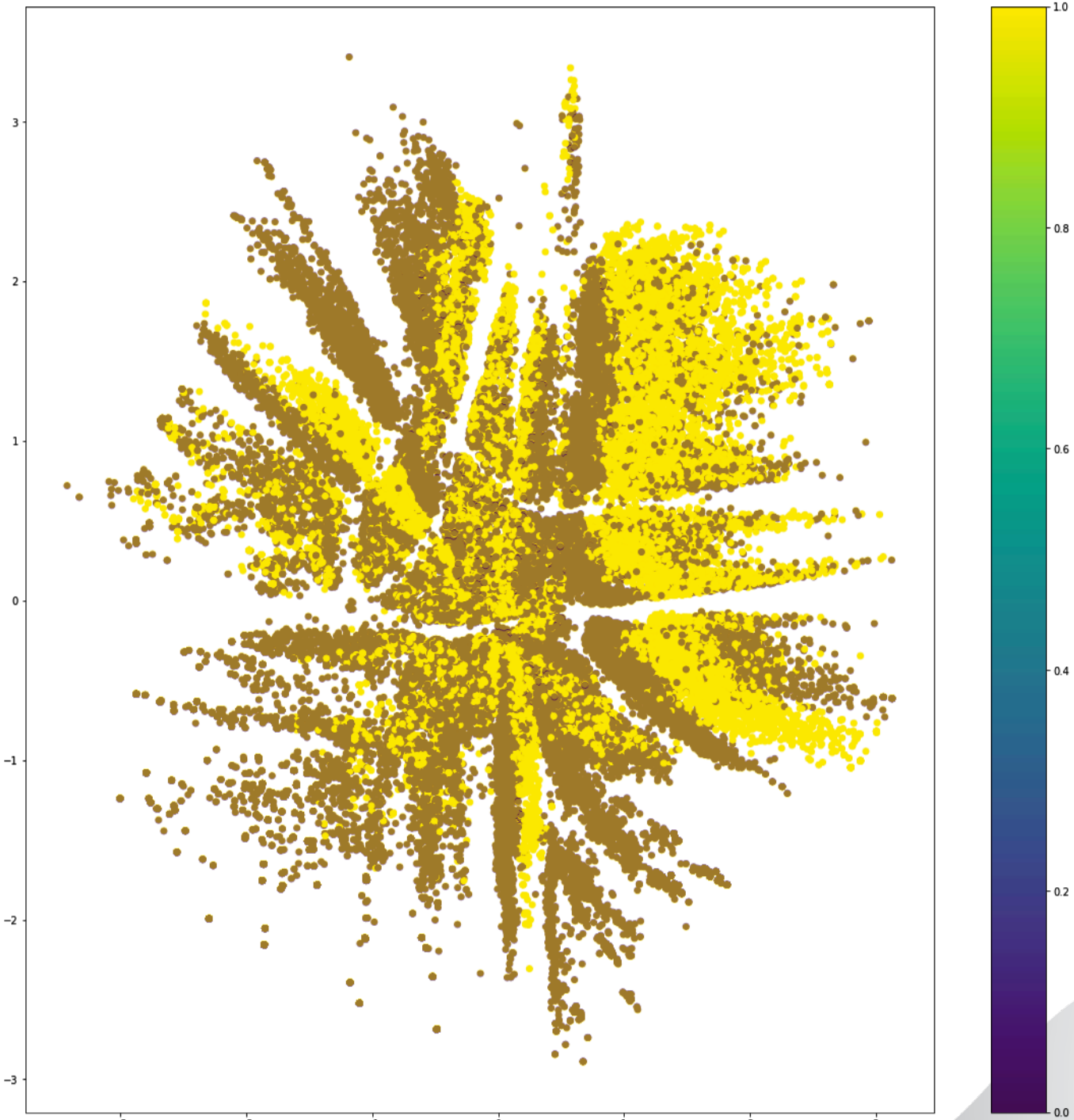

TREND MICRO™

Public




TREND MICRO™

Top500
+
Public

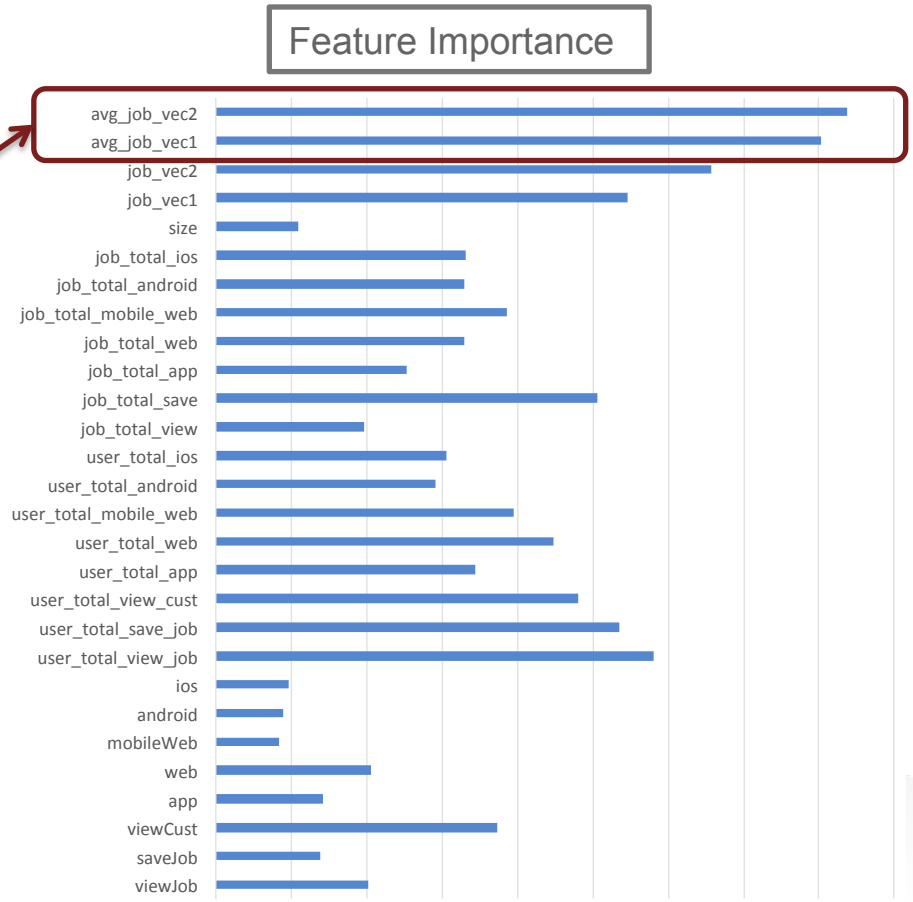



TREND MICRO™

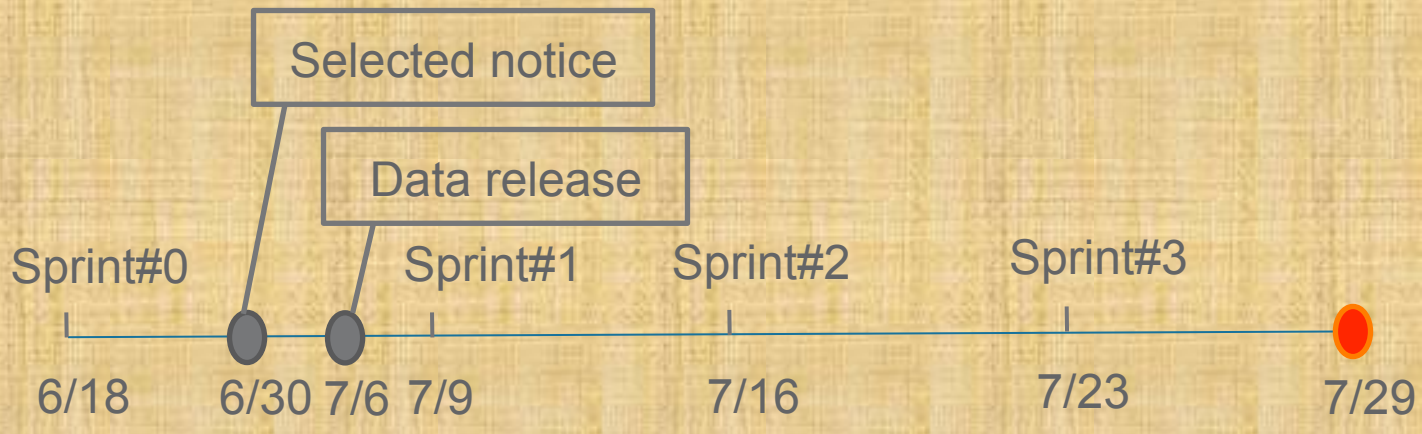
# User-Job feature vector

- User's browsed jobs → Averaging job
- Using user behavior log to get some features

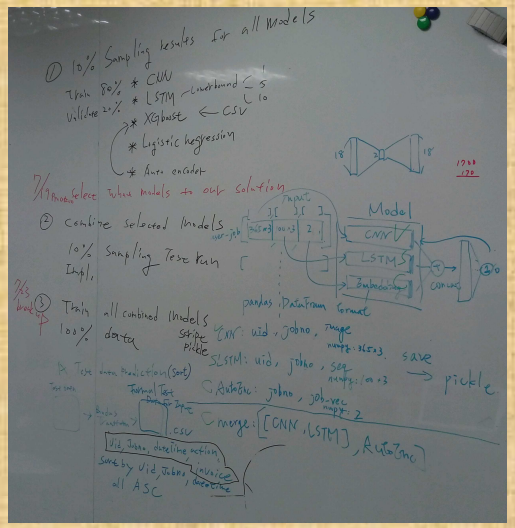
User's browsed job set = $\{Job_1, Job_2, \ldots, Job_i\}$
avg_job_vec = $Sum(Job_{set})/count(Job_{set})$



# Timeline <Sprint#2>




Prepare Environment
Data Exploration



Data exploration
Feature extraction

Train models

- CNN
- LSTM
- MLP + VAE
- All-in-one

Hyper-parameter and ensemble models

2nd

AWS

TREND MICRO

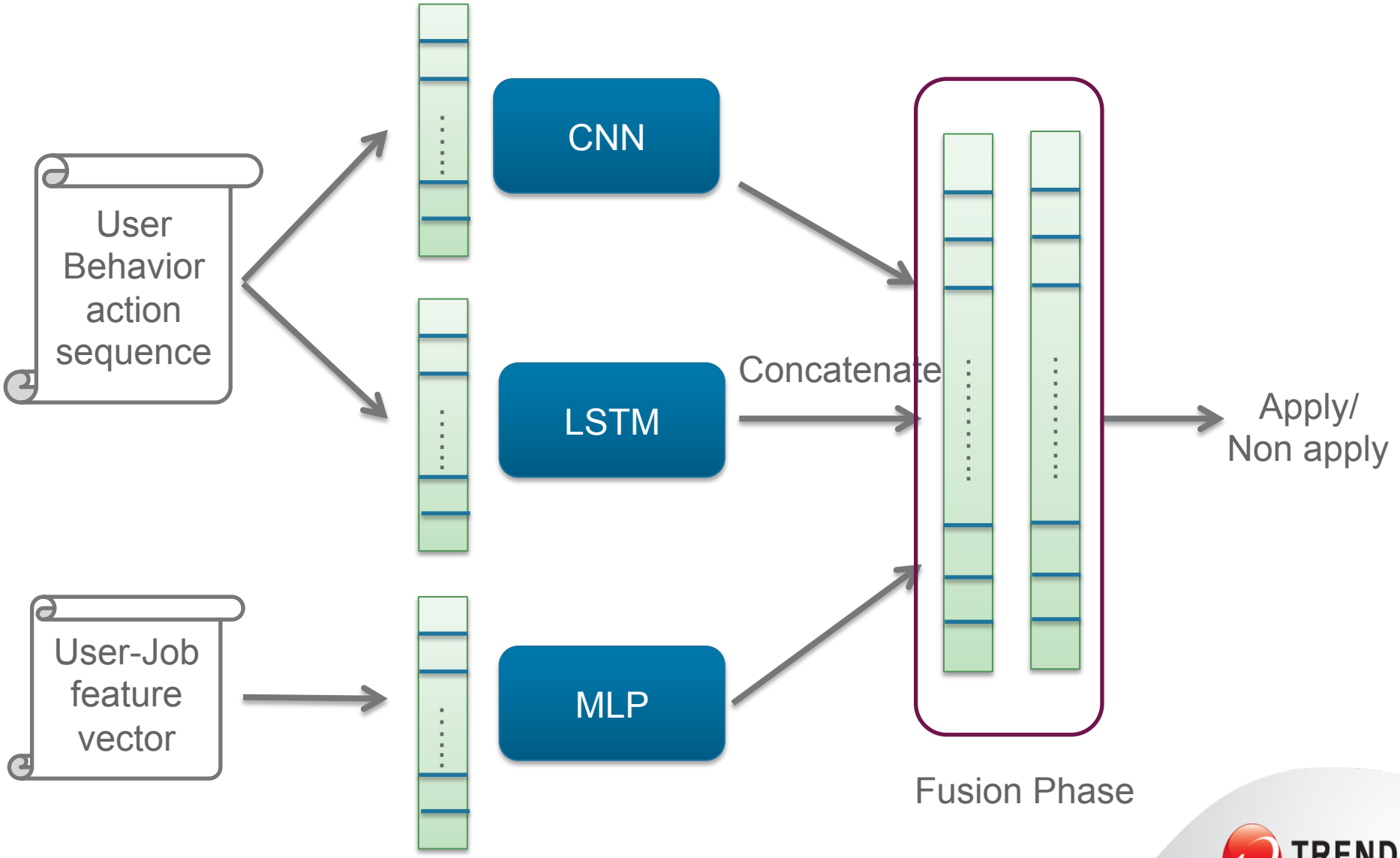
# Data Clean/Sampling

- Filter Noise (total action count = 1)
- Split balanced Training/Testing Data (8:2)
- Final Model version
  - Using class weight to against unbalanced data

# Modeling

- Convolutional neural network (CNN)
- Long Short-Term Memory neural network (LSTM)
- Multi-Layer Perceptron (MLP) + Variational AutoEncoder (VAE)
- All in One
  - CNN
  - LSTM
  - MLP + VAE

# All-in-One Model Architecture



# Result (balanced data)

- Training : Testing = 8:2
- Non-apply : Apply = 1:1

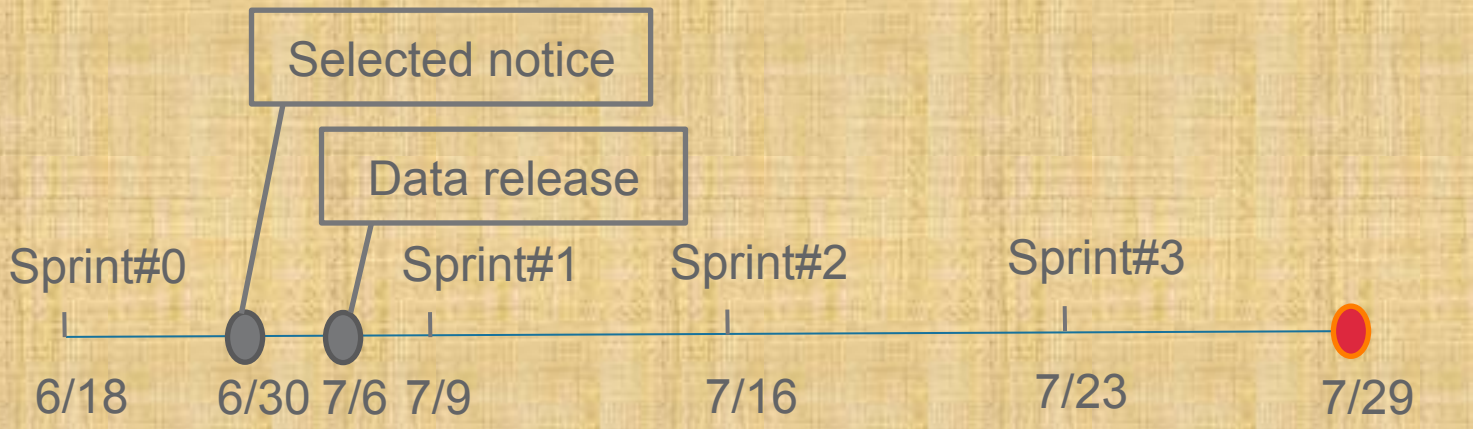
| Model | Accuracy |
| --- | --- |
| CNN | 57% |
| LSTM | 67.04% |
| MLP (VAE) | 68.2% |
| All-in-One | 69% |

# Result (unbalanced data)

- Training : Testing = 8:2
- Non-apply : Apply = 3:1

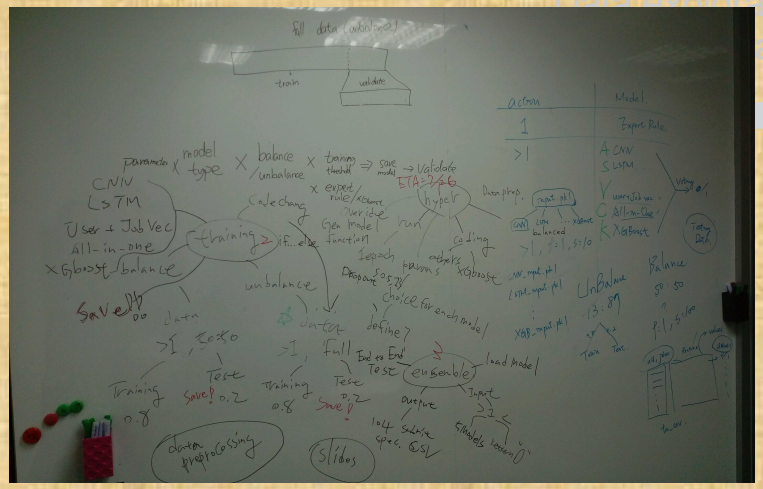
| Model | Accuracy |
| --- | --- |
| CNN | 74.6% (remove maxpooling) |
| LSTM | 70.3% |
| MLP (VAE) | 76% |
| All-in-One | 77.6% |

# Timeline <Sprint#3>




Prepare Environment
Data Exploration

Data exploration
action



Train models

- CNN
- LSTM
- MLP + VAE
- All-in-one

Hyper-parameter and ensemble models

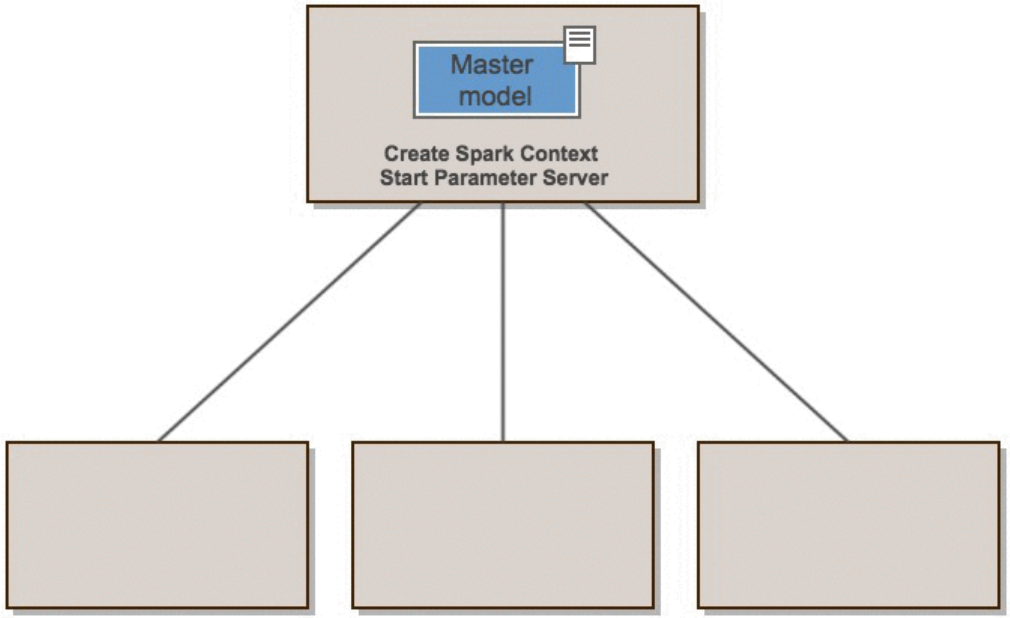
2nd

AWS

TREND MICRO

# Distributed Deep Learning with Keras & Spark

- Two-level distribution
  - Distributed Training Data (Elephas)
  - Distributed Model Parameter (Hyperas)



# Distributed Training Data (Elephas)

- Parallelize Training Data
  - --total-executor-cores
  - --executor-cores
  - numPartition

```
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.optimizers import SGD
model = Sequential()
model.add(Dense(128, input_dim=784))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(Dense(10))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer=SGD())
```

```
from elephas.utils.rdd_utils import to_simple_rdd
rdd = to_simple_rdd(sc, X_train, Y_train)

from elephas.spark_model import SparkModel
from elephas import optimizers as elephas_optimizers

adagrad = elephas_optimizers.Adagrad()
spark_model = SparkModel(sc,model, optimizer=adagrad, frequency='epoch', mode='asynchronous', num_workers=2)
spark_model.train(rdd, nb_epoch=20, batch_size=32, verbose=0, validation_split=0.1)
```

TREND MICRO

# Distributed Model Parameter (Hyperas)

- Deep architecture
  - Choice {64, 128, 256 …}
  - Uniform {0, 1}
  - Epochs
  - Batch Size
  - Layers
- Algorithm
  - Cluster
    - Random Search
  - Local
    - Tree of Parzen Estimators
    - Annealing

```
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation
from keras.optimizers import RMSprop

model = Sequential()
model.add(Dense(512, input_shape=(784,)))
model.add(Activation('relu'))
model.add(Dropout({{uniform(0, 1)}}))
model.add(Dense({{choice([256, 512, 1024])}}))
model.add(Activation('relu'))
model.add(Dropout({{uniform(0, 1)}}))
model.add(Dense(10))
model.add(Activation('softmax'))

rms = RMSprop()
model.compile(loss='categorical_crossentropy', optimizer=rms)

model.fit(X_train, Y_train,
          batch_size={{choice([64, 128])}},
          nb_epoch=1,
          show_accuracy=True,
          verbose=2,
          validation_data=(X_test, Y_test))
score, acc = model.evaluate(X_test, Y_test, show_accuracy=True, verb
print('Test accuracy:', acc)
return {'loss': -acc, 'status': STATUS_OK, 'model': model.to_yaml(),
```

TREND MICRO

# Ensemble (Hard Voting)

- CNN
- LSTM
- MLP (VAE)
- All-In-One

TREND MICRO

# Summary

- Action sequence
- Job hidden space
  - High feature importance
  - Increase the more user feature for user de-identification
- Ensemble models can cover more aspects
- Combine features from different views to correlate
  - Action sequence feature + user-job feature

# Thank You

TREND MICRO™

# Retrospective

- Do more on data exploration
  - Traditional ML
  - DL
- Centralized preprocessing procedure
- Agile end-to-end
- Minimize memory usage
  - Train on batch