[2021 Spring] Principles of Deep Learning Project

Twitter Recommendation System – Recsys challenge 2021

June 9, 2021

Team 4
AIGS 20215029 YoungIn Kwon
AIGS 20215113 JoonHyung Kim
IE 20161200 YeongHo Lee

CONTACT

Ulsan National Institute of Science and Technology
Address 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Korea
Tel. +82 52 217 0114 **Web.** www.unist.ac.kr

CONTENTS

Introduction: about competition

Competition objective
The **'real-world task'** of tweet engagement prediction **'in a dynamic environment'**.
※ **tweet engagement prediction**: Like, Retweet, Quote, and replies

Constraints
- **Latency** constraints (numerous dataset size)
- Satisfy **multi-goal optimization**, accuracy and fairness
- Targets are **divided as 5 groups per popularity for scoring fairness**

Problem Definition
Predict **0(non-interact)** or **1(interact)** label of different engagement type **(Binary Classification)**

Metric

- **Average Precision Score – Accuracy**

$$AP = \sum_{n} (R_n - R_{n-1}) P_n$$

- **Relative Cross Entropy(RCE) – Fairness**

$$RCE = \frac{CE_{naive} - CE_{pred}}{CE_{naive}} * 100$$

Introduction: dataset

Dataset served as Twitter APIs

- Train data is **259 numbers of TSV files** which have **3 million rows and 800MB** per one file
- It is **closed to 1 billion** data points
- Dataset **split as train and valid dataset by number of weeks** per a month
  ※ Week 1-3: train / Week 4: valid

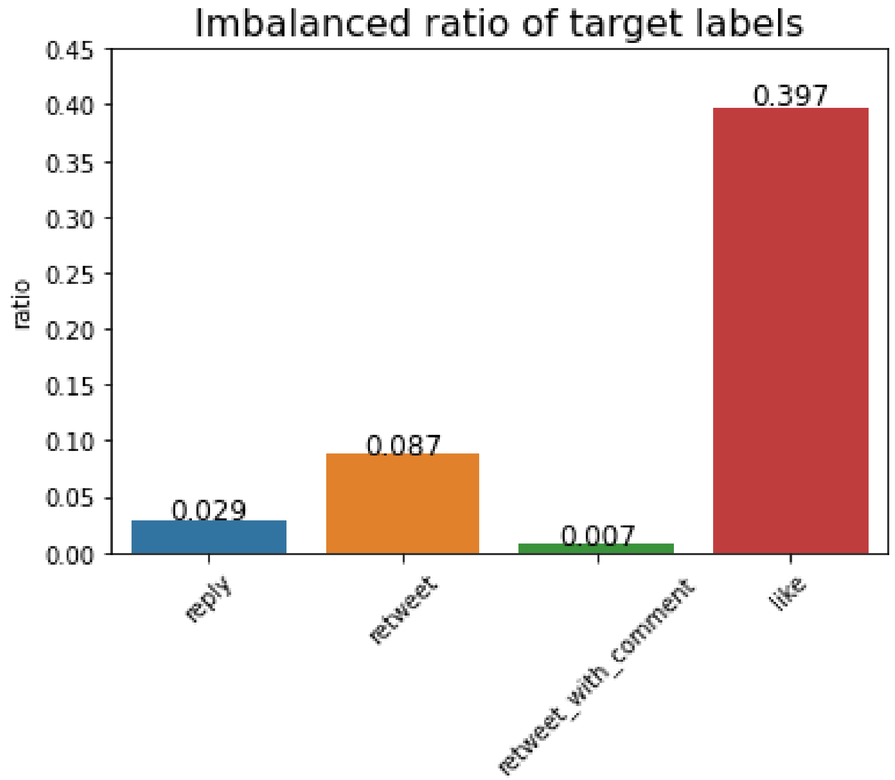
Columns of Dataset

Items	Names
Features	"text_ tokens", "hashtags", "tweet_id", "present_media", "present_links", "present_domains","tweet_type","language", "tweet_timestamp", "engaged_with_user_id", "engaged_with_user_follower_count", "engaged_with_user_following_count", "engaged_with_user_is_verified", "engaged_with_user_account_creation","enaging_user_id", "enaging_user_follower_count", "enaging_user_following_count", "enaging_user_is_verified“,"enaging_user_account_creation", "engagee_follows_engager"
Labels	"reply_timestamp", "retweet_timestamp", "retweet_with_comment_timestamp", "like_timestamp"

Introduction: Restrictions

Dataset suffered imbalance at all of labels

- Among them, 3 labels have severe imbalance ("reply_timestamp", "retweet_timestamp", "retweet_with_comment_timestamp")



Submission Environment

- e2-highmem-16 on GCP with 64GB memory
- 24 hours time limit
- Test dataset ~ 10 million rows

Machine Name	vCPUs[1]	Memory (GB)	Max number of persistent disks (PDs)[2]	Max total PD size (TB)	Local SSD	Maximum egress bandwidth (Gbps)[3]
e2-highmem-2	2	16	128	257	No	4
e2-highmem-4	4	32	128	257	No	8
e2-highmem-8	8	64	128	257	No	16
e2-highmem-16	16	128	128	257	No	16

Methodologies: text token

Text token is tokenized text of target tweet post

For utilizing token, it needs to be converted as arbitrary feature by some methods.

There are methodologies adopted below:

	tweet_id	text_ tokens
0	D6621E1038904DA83CBBA1DE9F4FFA7A	101 56898 137 14657 11462 11460 66730 131 1139...
1	1F0C624B6B3455AA8C14A7C4EF6B342E	101 56898 137 52544 10147 12396 11233 11975 13...
2	F944E479EBDEEECBFBA03F47D8B5B79A	101 56991 216 216 19318 11301 14120 131 120 12...
3	0F98BD50C159E189E7F6F8203227FC36	101 56898 137 18087 11205 11090 11010 10269 13...
4	3C1A2B662FBA0436DCBCCD488B08E2D4	101 56898 137 12882 31604 10291 89525 11359 11...

Method	Model type	Output
TFIDF-GloVe	Unsupervised	d dimension vector
SIF(Embedding by GloVe)	Unsupervised	d dimension vector
DAN(Deep Average Network)	Supervised	d dimension vector
DEC(Deep Embedded Clustering)	Unsupervised	n clustered integer

Methodologies: vectorizing by GloVe[1]

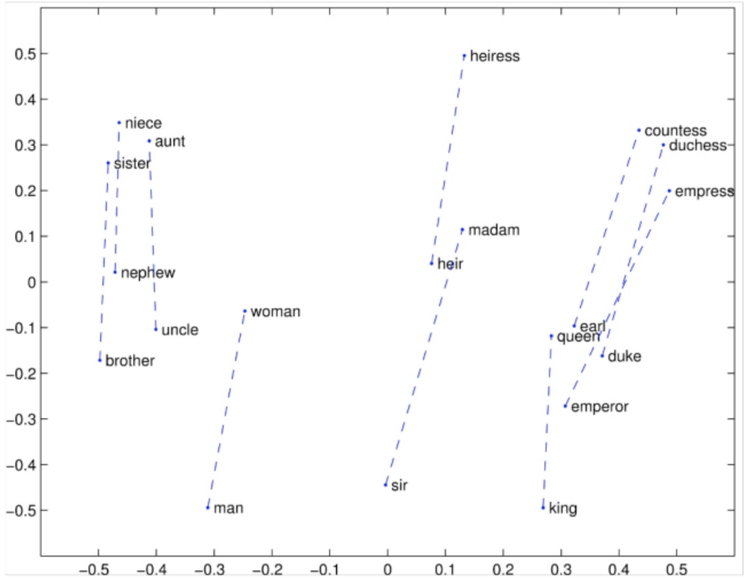
There is word embedding as background of implementation

GloVe model selected

It emerged for covering drawback of LSA and Word2Vec model
- LSA(Latent Semantic Analysis): weakness at analogy task
- Word2Vec: weakness of locality from window

It adopts Co-occurrence Probability made from Co-occurrence matrix

The objective is finding function F given three target vector like below:



$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Methodologies: weighted average (tfidf-GloVe, SIF[2])

The simplest implementation is to take averages about all vectors of sentence

However, it could be lost some sense of important word by concatenating

Then, try “Weighted Average” by tf-idf or SIF(SIMPLE BUT TOUGH-TO-BEAT) method

$$TF-IDF = tf(d,t) * idf(t)$$
$$= tf(d,t) * log(\frac{n}{df(t)})$$

- $tf(d,t)$ number of counts the word t appeared in specific document d,
- $df(t)$ inverse of number of counts the word t appeared in whole documents

TF-IDF formula

Algorithm 1 Sentence Embedding

Input: Word embeddings $\{v_w : w \in \mathcal{V}\}$, a set of sentences $\mathcal{S}$, parameter a and estimated probabilities $\{p(w) : w \in \mathcal{V}\}$ of the words.
Output: Sentence embeddings $\{v_s : s \in \mathcal{S}\}$

1: **for all** sentence s in $\mathcal{S}$ **do**
2: $v_s \leftarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{a+p(w)} v_w$
3: **end for**
4: Form a matrix X whose columns are $\{v_s : s \in \mathcal{S}\}$, and let u be its first singular vector
5: **for all** sentence s in $\mathcal{S}$ **do**
6: $v_s \leftarrow v_s - uu^\top v_s$
7: **end for**

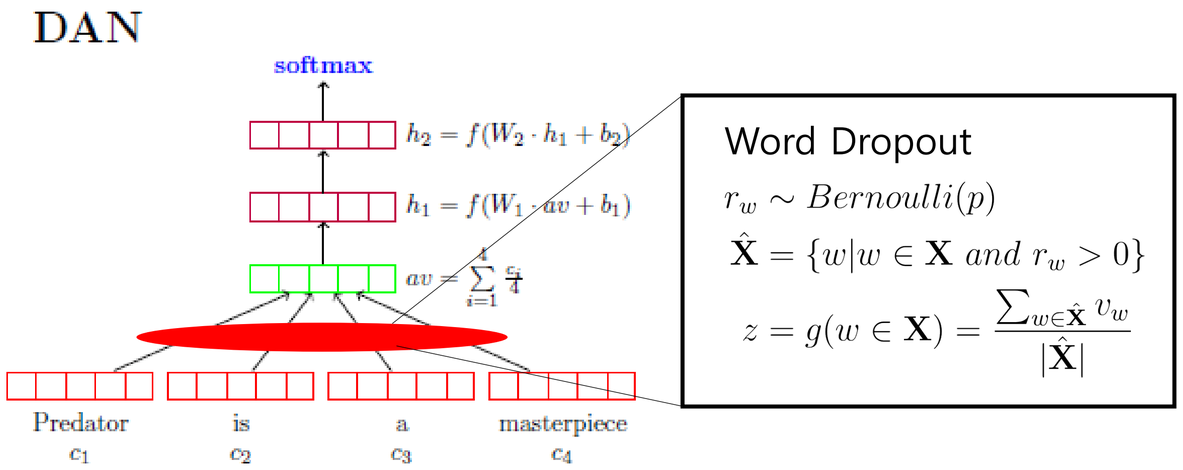
SIF Pseudo Code

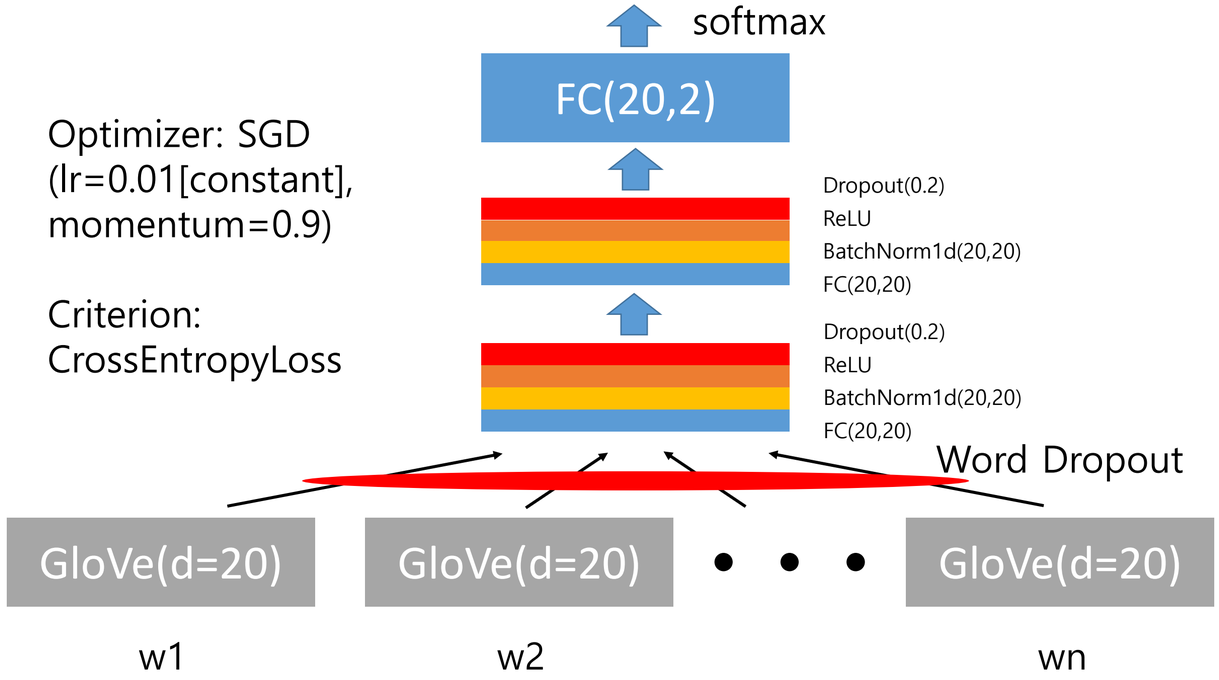
Methodologies: DAN (Deep Averaging Network[3])

For comparing with unsupervised average method, DAN(Deep Average Network) used

DAN is learned by 2 fully connected layer using averaged word vector input values

To prevent overfitting, word dropout utilized before averaging word embeddings



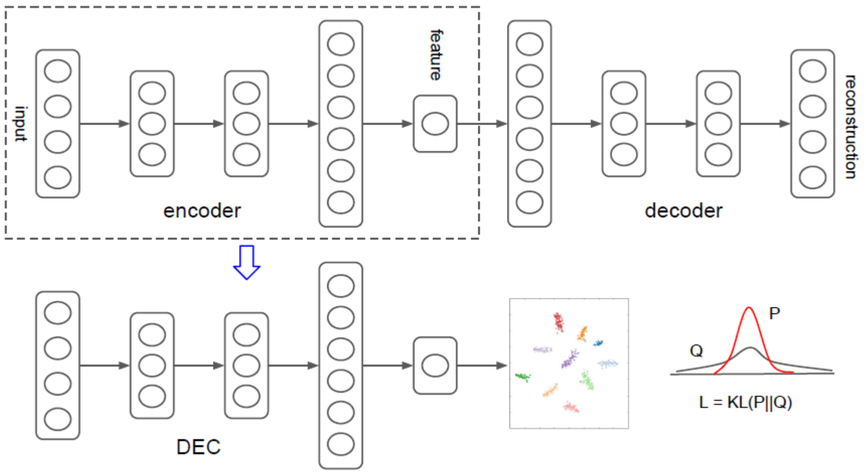


Methodologies: DEC (Deep Embedded Clustering[4])

Unlike vectorization method, another method is utilizing clustered feature

For this, DEC(Deep Embedded Clustering) which allocates cluster by pretrained encoder

It proceeded 2 phases greedy layer wised pretrain and improve clustering by pretrained encoder



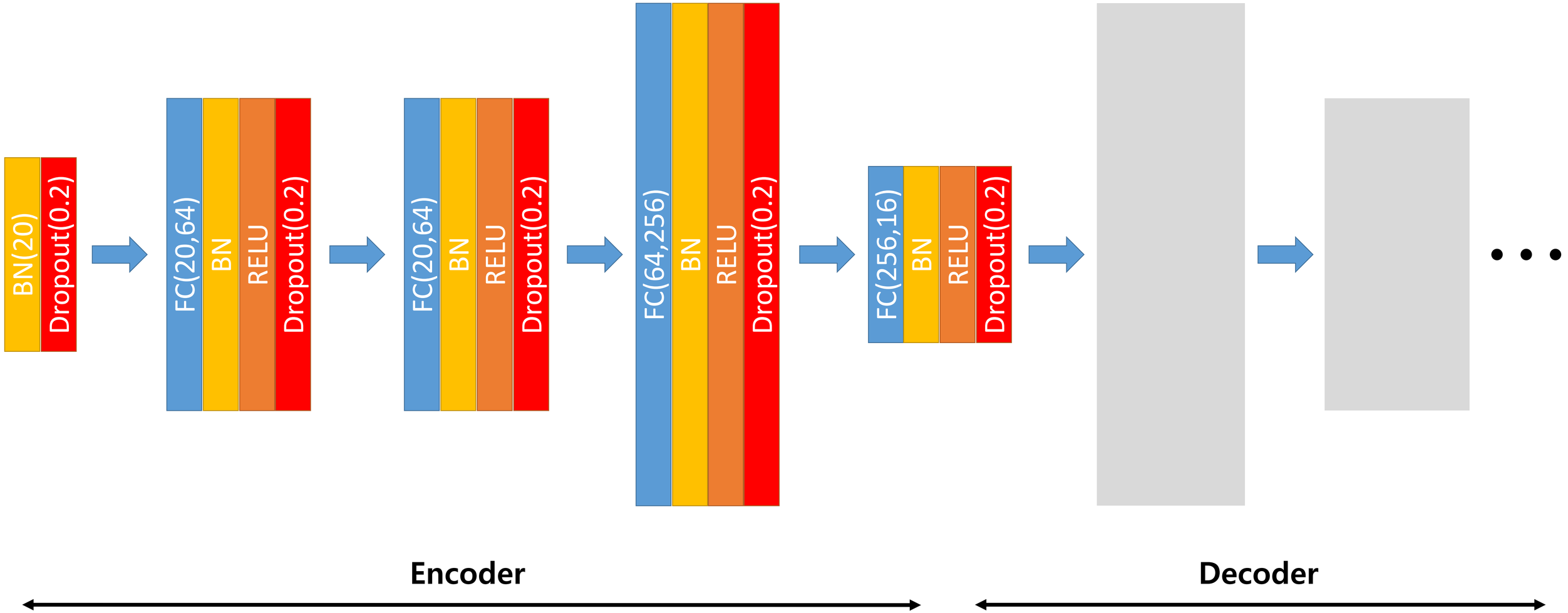
Soft Assignment

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1 + ||z_i - \mu_{j'}||^2/\alpha)^{-\frac{\alpha+1}{2}}}$$

Iteratively refine cluster by KL Divergence Minimization

$$L = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$$

Methodologies: DEC (Deep Embedded Clustering[4])



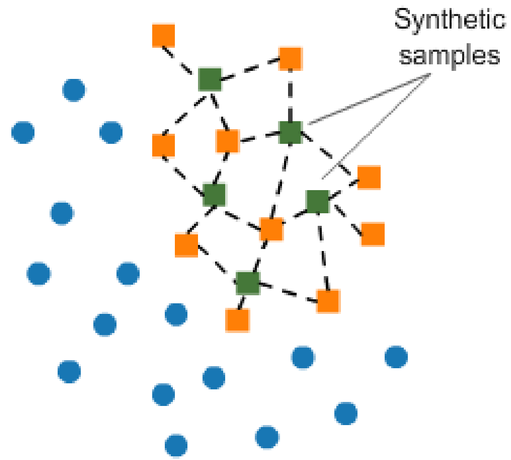
Methodology：Generative Adversarial Network for Data Imbalance

Extremely imbalanced data[5]

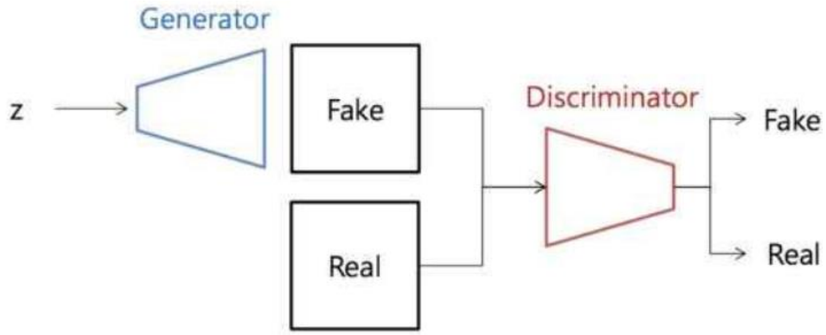
Re-extraction：It's simple, but Overfitting happens

SMOTE : Overfitting mitigates, but generates potential noise data

GAN : Troubleshooting Potential Noise Data Generation



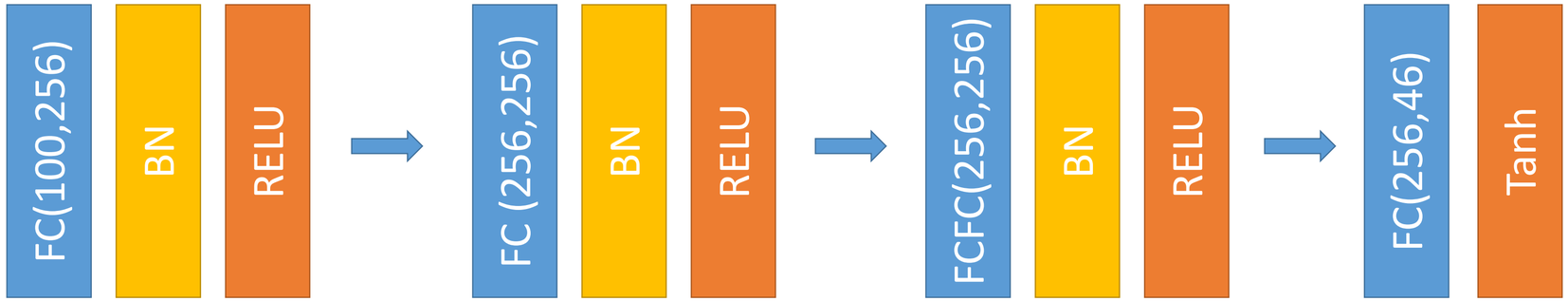
SMOTE



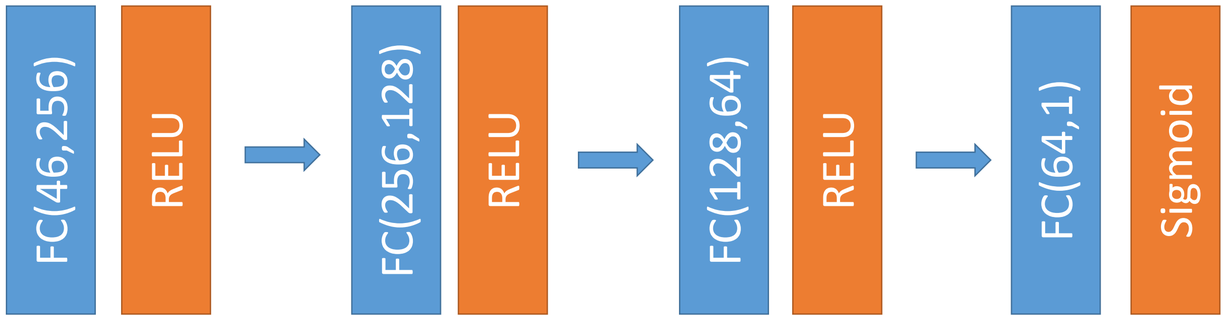
GAN

FIRST IN CHANGE

Methodology：Generative Adversarial Network for Data Imbalance



Generator Model



Discriminator Model

Criterion : BCELoss
Optimizer : Adam

FIRST IN CHANGE

Methodology : Classification model - ANN

Binary classification

Decision Tree 2:17 Neural Network

Multiclass classification

Decision Tree 0:7 Neural Network

Regression

Decision Tree 4:12 Neural Network

Compared on OpenML datasets[6]

It performs better than other machine learning models

FIRST IN CHANGE

Methodology : ANN

Continuous variable : engaged_with_user_following_count, engaged_with_user_follower_count, engaging_user_follower_count, engaging_user_following_count, engaged_time - engaging_time tweet_time - engaging_time, tweet_hour, language(Rank), num_GIF, num_Video, num_Photo, engaged_time, engaging_time, tweet_timestamp, ratio_enaging_follow, ratio_enagaged_follow

-> **RobustScaler scaler**

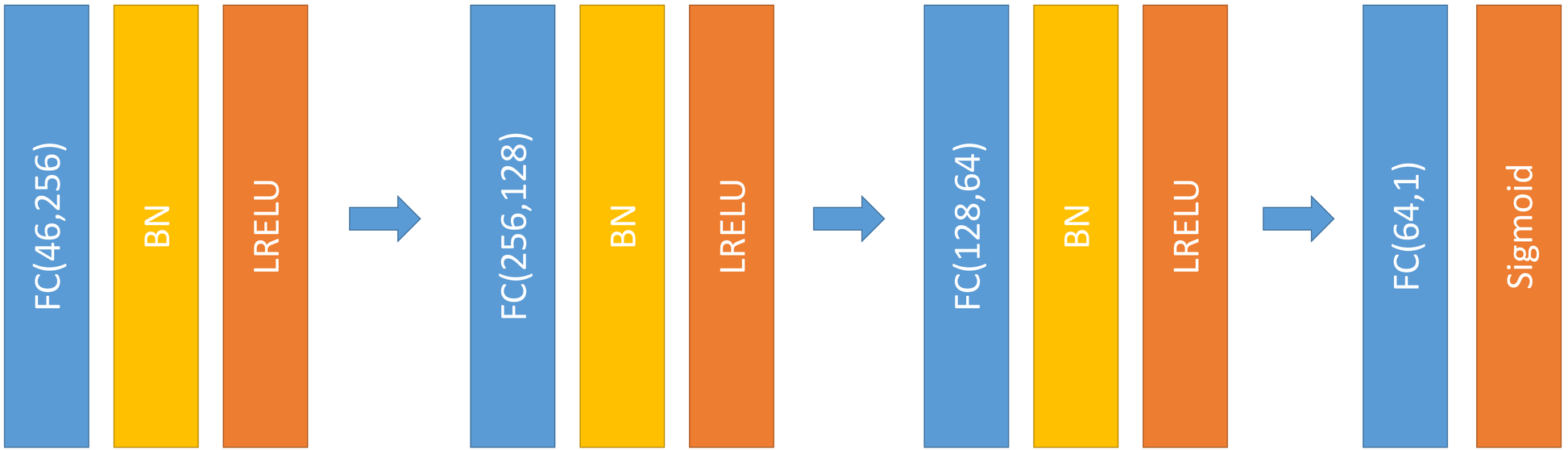
Discrete variable(Dummy variable) : tweet_type_0, … ,tweet_type_3, engaged_with_user_is_verified, engaging_user_is_verified, engagee_follows_engager, present_links, bool_(engaged_time > engaging_time)

Text variable (SIF, TFIDF, DAN) : sentenc_vector_0, … , sentenc_vector_19, cluseter

Input data has 46 features

FIRST IN CHANGE

Methodology : ANN



ANN Model

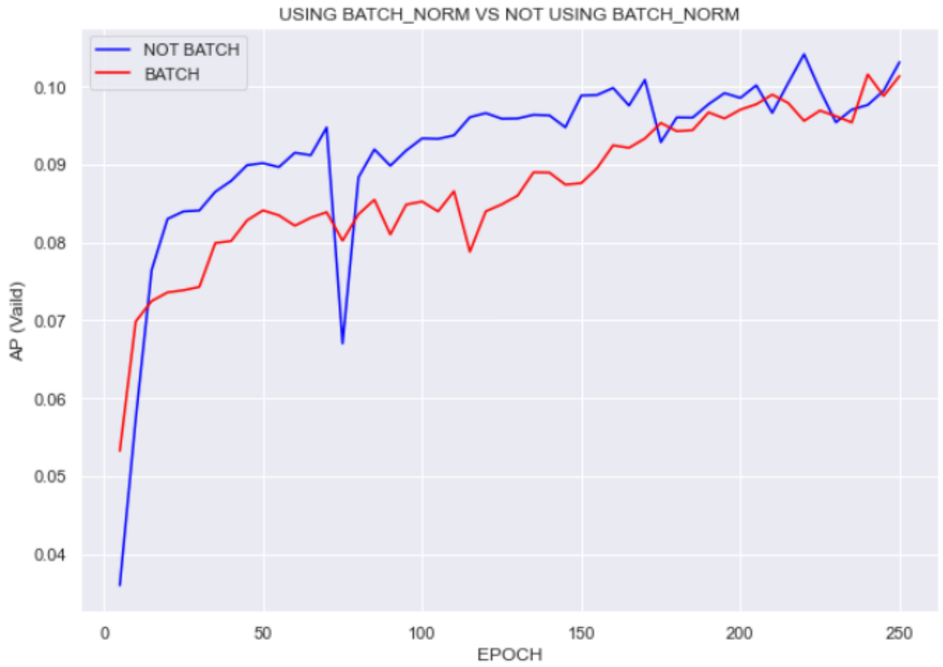
Criterion : SmoothL1Loss
Optimizer : Adam

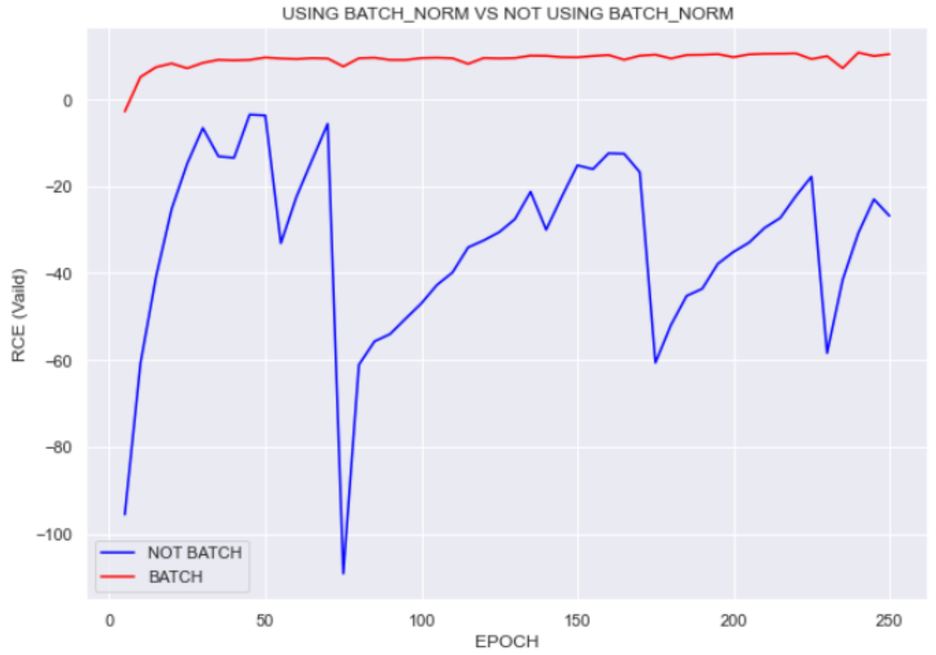
FIRST IN CHANGE

Experiments

Valid Set: **Extract 50% randomly from Test Data**

Experiments : Batch Normalization (Not Using Text token)

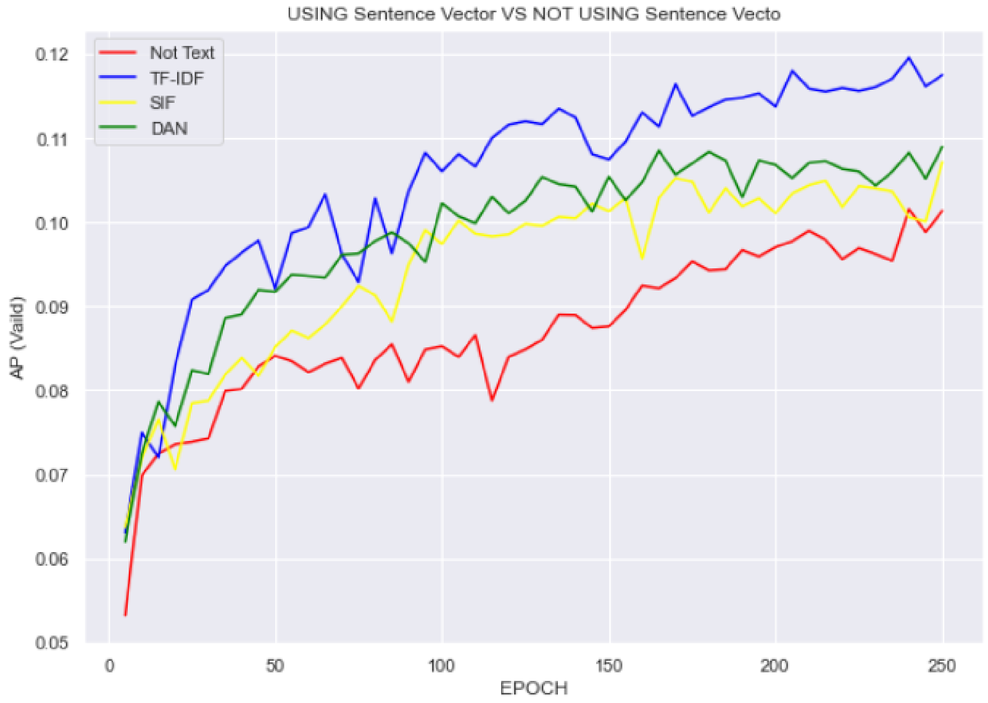


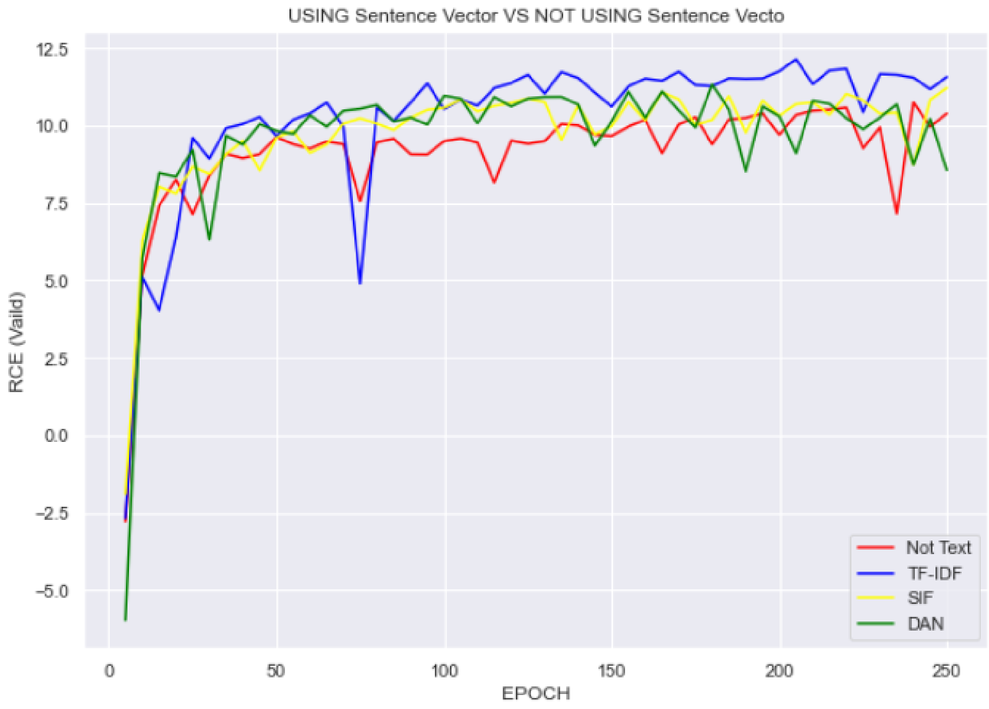


Baseline Model : **Using Batch Normalization**

FIRST IN CHANGE

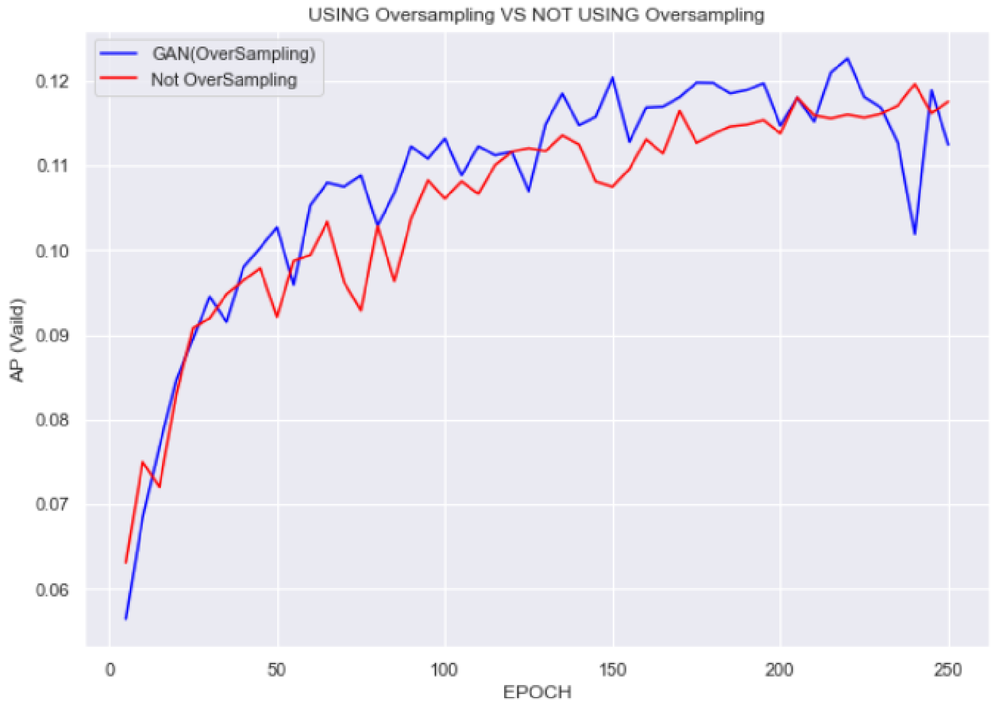
Experiments : Sentence Vector

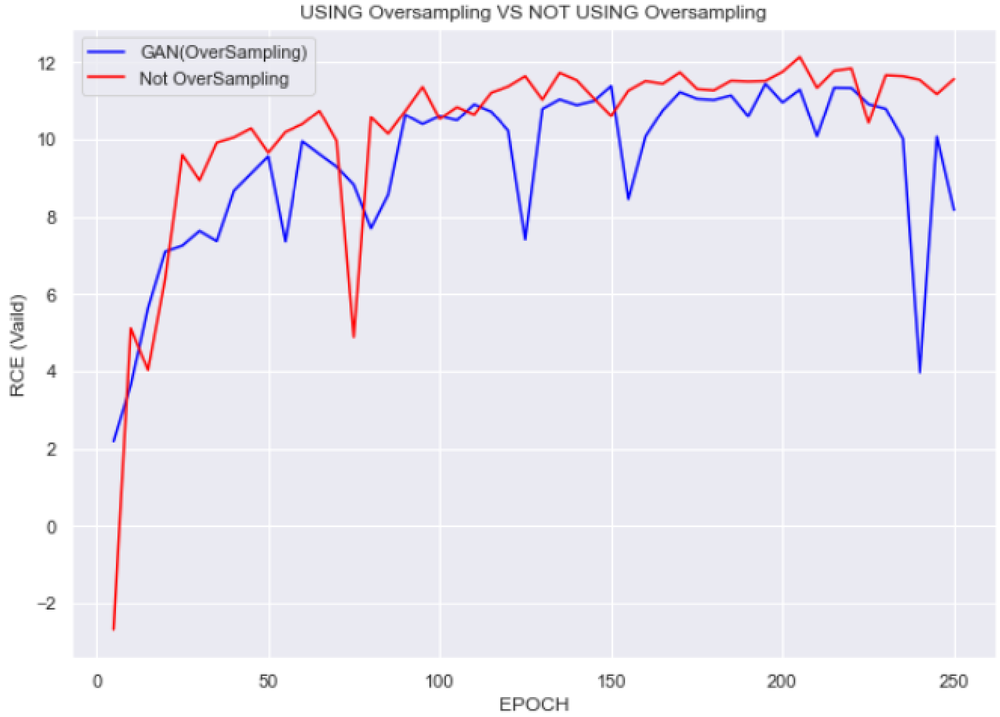




Baseline Sentence Vector : **tfidf-GloVe**

Experiments : OverSampling (GAN)





Experiment

Batch Norm	Sentence	GAN	AP	RCE
X	X	X	0.104171	-3.5193
O	X	X	0.10158	10.7507
O	SIF	X	0.10718	11.2185
O	DAN	X	0.10898	11.3233
O	tfidf-GloVe	X	0.11961	12.1408
O	tfidf-GloVe	O	0.12263	11.4452

FIRST IN CHANGE

Conclusion

1. Text is important to followers on Twitter
   (Approximately 20% increase in performance **(tfidf-GloVe)**)
   -> Followers read and sympathize

2. Batch normalization is more effective for **RCE** than **AP**
   -> Batch normalization is over performing the naive prediction[7]

3. **GAN** is more effective for **AP** than **RCE**

4. Model will **be split for 5 groups by popularity** for enhancing **RCE** further

To increase **RCE**, we are going to create a model that takes into account followers numbers.

Reference

[1] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
[2] Arora, Sanjeev, Yingyu Liang, and Tengyu Ma. "A simple but tough-to-beat baseline for sentence embeddings." (2016).
[3] Iyyer, Mohit, et al. "Deep unordered composition rivals syntactic methods for text classification." Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers). 2015.
[4] Xie, Junyuan, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis." International conference on machine learning. PMLR, 2016.
[5] 정성욱, 김이슬, 김일곤, 임경식. GAN을 이용한 데이터 불균형 문제에 관한 연구. 한국통신학회 학술대회논문집. 2019;():1390-1391.
[6] “Decision Tree vs Neural Network”. mljar.com. https://mljar.com/machine-learning/decision-tree-vs-neural-network
[7] Luca Belli, et al “Privacy-Aware Recommender Systems Challenge on Twitter’s Home Timeline.” (2020)

THANK YOU