Phishing Site Detection
Machine Learning

BOB 보안제품개발9기
임하늘

<목차>

1절) 연구 개발 수행 내용

Machine Learning을 이용하여 phishing site Detection으로

총 2가지항목에 맞추어 진행했다

1). Phishing site 분석

2). Supervised learning

1. Deep Learning (CNN1D, CNN+LSTM, LSTM) 문자 처리 기반
2. Machine Learning (Tokenized, pipeline (logistic regression)
3. Domain Entropy (Random Forest, Logistic regression, Decision Tree)
4. DomainEntropy + request

   (Random Forest, Logistic regression, Decision Tree)

3). Unsupervised learning

1. PCA(주성분 분석)(Unsupervised learning)

2절) 연구결과

Phishing site는 특징으로 자신의 목적을 숨기기 위해 URL를 비이상적으로 길게 만들어 자신의 목적을 숨기는 경우 URL Entropy를 구하여 각각 URL에 엔트로피 치수를 보고 탐지 결과를 보여줬으나 asq, bitily 같이 URL를 짧게 만들게 되면 phishing site의 URL특징점이 사라져 오탐률이 높아지는 한계점을 보였다.

2-1). Phishing site 데이터 수집

Phishing site data 수집은 OpenPhishing + PhishingTank + Kaggle

1. OpenPhishing(https://openphish.com/phishing_feeds.html)

(자동화된 독립형 플랫폼 피싱사이트를 식별하고 사람의 개입없이 실시간으로 인텔리전스 분석수행 5000개 phishinURL제공)

http://tahani2017.com/css/hh
https://gainulkkhwghtecdchvuypjswb-dot-gloff00403993445.uk.r.appspot.com/?utm_source=sendgrid.com&utm_medium=email&utm_campaign=website
https://bgsocbvdriwyogaixnkuxbsjcv-dot-gloff00403993445.uk.r.appspot.com/
https://gmahxwwmltrzsvcrclucqpkewh-dot-gloff00403993445.uk.r.appspot.com/?utm_medium=email&utm_campaign=website&utm_source=sendgrid.com
https://notags.fbookcon-lrypouwo.electronicamarvan.com/nextVRF.html?location=e8dbeba70ced8ddf68b4be751e024eb03
http://loginmeabernewshare.gq/
https://loginmeabernewshare.gq/
https://utyraxdogvulbqsmmtwdolldvp-dot-gloff00403993445.uk.r.appspot.com/?utm_campaign=website&utm_source=sendgrid.com&utm_medium=email
https://ghpbpbwvjcaizndgvhxfaganiq-dot-gloff00403993445.uk.r.appspot.com/
https://vqyxllvvnmbpardczrpqkbjrua-dot-gloff00403993445.uk.r.appspot.com/
https://ovdmfdhkdotfzngpxwizumhxgd-dot-gloff00403993445.uk.r.appspot.com/?utm_source=sendgrid.com&utm_medium=email&utm_campaign=website
https://yuyjawyrptvwwnyitddnwvjztx-dot-gloff00403993445.uk.r.appspot.com/?utm_medium=email&utm_campaign=website&utm_source=sendgrid.com
https://vzperrwqqfpnygaceojlaoyzfh-dot-gloff00403993445.uk.r.appspot.com/?utm_campaign=website&utm_source=sendgrid.com&utm_medium=email
https://auvgwpxjrzqvqiwuzkcbqtomwz-dot-gloff00403993445.uk.r.appspot.com/?utm_campaign=website&utm_source=sendgrid.com&utm_medium=email
https://bpztbwcnehrjajooznwtgvdbjh-dot-gloff00403993445.uk.r.appspot.com/?utm_source=sendgrid.com&utm_medium=email&utm_campaign=website
https://mtcdfhybsvbkshoadxpypknkgk-dot-gloff00403993445.uk.r.appspot.com/?utm_campaign=website&utm_source=sendgrid.com&utm_medium=email
https://aqyjpsnghntierirlpxnkplmhm-dot-gloff00403993445.uk.r.appspot.com/?utm_campaign=website&utm_source=sendgrid.com&utm_medium=email
https://unoozabunxjrelujydhvapxesd-dot-gloff00403993445.uk.r.appspot.com/?utm_source=sendgrid.com&utm_medium=email&utm_campaign=website

2. PhishTank(https://www.phishtank.com/)

(PhishTank는 인터넷에서 피싱에 대한 데이터 및 정보를 제공하는 협업 정보 센터. 또한 PhishTank는 개발자와 연구원이 무료로 안티 피싱 데이터를 애플리케이션에 통합할 수 있는 개방형 API를 제공.)

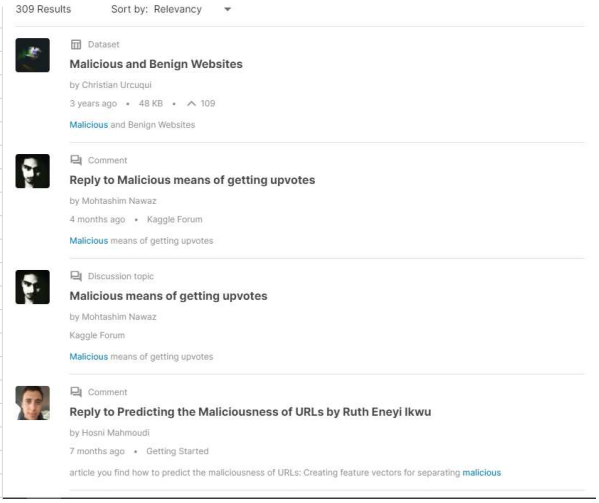
신분증	URL
6865116	http://vfo1.talk4fun.net/voq
6865115	https : //atlbha.com.sa/2/secure.bus1ness.bt.c0m/ind ...
6865114	https : //mail-confirmation-process.glitch.me/? email ...
6865113	https://droit-interim.com/bnps/ntbn.html
6865112	https : //ruralserviciosdeahorasbancooperativos.laga ...
6865111	https://amalicjkda.top/pc/
6865109	https : //centerspan.com/login/office365f/office365f ...
6865108	http://down.slothy.in/
6865107	https://instructpayee.com/halifax/
6865106	https://secure-lloyd-payee.net/
6865105	https : //itau.gq/~/index.html? IEBeIeLh5YnRTMP8Zni8L ...
6865104	http://post-e.top/hly.html
6865103	https://www.ellatinodigital.com/wp-content/inv/z/
6865101	https : //d-service-bae34.web.app/connecting.html? al ...
6865100	https://bcc.kevinheadley.com/ndbn/rndex.php

phish_id	url	phish_detail_url	submission_verified	verification_online	target
6865115	https://atlbha.com.sa/2/secure.bus1ness.bt.c0m/index.htm	http://www.phishtank.com/phish_detail.php?phish_id=6865115	2020-11-2 yes	2020-11-2 yes	Other
6865114	https://mail-confirmation-process.glitch.me/?email#abuse@optusnet.com	http://www.phishtank.com/phish_detail.php?phish_id=6865114	2020-11-2 yes	2020-11-2 yes	Other
6865111	https://amalicjkda.top/pc/	http://www.phishtank.com/phish_detail.php?phish_id=6865111	2020-11-2 yes	2020-11-2 yes	Amazon.com
6865109	https://centerspan.com/login/office365f/office365files/confidential/	http://www.phishtank.com/phish_detail.php?phish_id=6865109	2020-11-2 yes	2020-11-2 yes	Other
6865108	http://down.slothy.in/	http://www.phishtank.com/phish_detail.php?phish_id=6865108	2020-11-2 yes	2020-11-2 yes	eBay, Inc.
6865107	https://instructpayee.com/halifax/	http://www.phishtank.com/phish_detail.php?phish_id=6865107	2020-11-2 yes	2020-11-2 yes	Other
6865106	https://secure-lloyd-payee.net/	http://www.phishtank.com/phish_detail.php?phish_id=6865106	2020-11-2 yes	2020-11-2 yes	Other
6865105	https://itau.gq/~/index.html?IEBeIeLh5YnRTMP8Zni8LnUduxGGUaJ8gY8t	http://www.phishtank.com/phish_detail.php?phish_id=6865105	2020-11-2 yes	2020-11-2 yes	Other
6865104	http://post-e.top/hly.html	http://www.phishtank.com/phish_detail.php?phish_id=6865104	2020-11-2 yes	2020-11-2 yes	Other
6865096	http://www.gestorlamanzaneque.com/bnps/ntbn.html	http://www.phishtank.com/phish_detail.php?phish_id=6865096	2020-11-2 yes	2020-11-2 yes	Other
6865086	https://xn--privnot-xs4c.com/	http://www.phishtank.com/phish_detail.php?phish_id=6865086	2020-11-2 yes	2020-11-2 yes	Other
6865085	https://xn--prvnots-8ya.com/	http://www.phishtank.com/phish_detail.php?phish_id=6865085	2020-11-2 yes	2020-11-2 yes	Other
6865084	https://xn--plvnote-953c.com/	http://www.phishtank.com/phish_detail.php?phish_id=6865084	2020-11-2 yes	2020-11-2 yes	Other
6865083	https://xn--plvnote-znb.com/	http://www.phishtank.com/phish_detail.php?phish_id=6865083	2020-11-2 yes	2020-11-2 yes	Other
6865082	https://xn--prvnot-gva1a.com/	http://www.phishtank.com/phish_detail.php?phish_id=6865082	2020-11-2 yes	2020-11-2 yes	Other
6865081	http://post-e.top/hly.html	http://www.phishtank.com/phish_detail.php?phish_id=6865081	2020-11-2 yes	2020-11-2 yes	Other
6865069	https://i-change.fun/	http://www.phishtank.com/phish_detail.php?phish_id=6865069	2020-11-2 yes	2020-11-2 yes	Other
6865068	https://netex24.cloud/	http://www.phishtank.com/phish_detail.php?phish_id=6865068	2020-11-2 yes	2020-11-2 yes	Other
6865067	https://cash-365.us/	http://www.phishtank.com/phish_detail.php?phish_id=6865067	2020-11-2 yes	2020-11-2 yes	Other
6865066	https://qiuckchange.biz/	http://www.phishtank.com/phish_detail.php?phish_id=6865066	2020-11-2 yes	2020-11-2 yes	Other
6865064	https://cash-i365.store/	http://www.phishtank.com/phish_detail.php?phish_id=6865064	2020-11-2 yes	2020-11-2 yes	Other
6865058	https://besichenge.site/	http://www.phishtank.com/phish_detail.php?phish_id=6865058	2020-11-2 yes	2020-11-2 yes	Other
6865057	https://kentiumiluminacion.000webhostapp.com/indexfile.php?email=aa	http://www.phishtank.com/phish_detail.php?phish_id=6865057	2020-11-2 yes	2020-11-2 yes	Other

3. Kaggle(https://www.kaggle.com/)

(2010년에 설립된 예측 모델 및 분석 대회플랫폼 다양한 데이터 분포 기업에서 데이터와 해결 과제를 주면 데이터 과학자들이 이를 해결하고 모델 개발 및 경쟁 현재 구글에 2017년 3월 인수되었음

http://lctdesign.com.sg/media/contacts/ima	bad
http://updatedoffice.xyz/alibaba.pl	bad
https://tergab2018.000webhostapp.com/apri	bad
https://itaoferencimentos.com/cadastro/inici	bad
https://www.gmailmsg.com/signin?t=eyJhbG	bad
http://steklomir.org/wp-content/uploads/tw	bad
http://frienbe.com/.ch/1	bad
http://frienbe.com/.ch/1/	bad
http://www.coolcoolers.com.au/bmo/b7f235	bad
https://situspokerjawa.org/ty/sw/index.php	bad
http://frienbe.com/.ch/1/details.html	bad
https://bitly.com/2EinMEO	bad
https://avaksystems.com/rover/9/XUbHIBMv	bad
https://avaksystems.com/rover/9/NhAgK9fYb	bad
https://avaksystems.com/rover/9/oFTplpYRn	bad
https://avaksystems.com/rover/9/5zRQTthI3	bad
https://avaksystems.com/rover/9/H6qhfsppz	bad
https://avaksystems.com/rover/9/eMbYP53Lx	bad
https://avaksystems.com/rover/9/HkndYKlDq	bad
https://avaksystems.com/rover/9/nSiKQu5JZ	bad
http://www.netbb-online.com/	bad



4. https://github.com/mitchellkrogza/Phishing.Database

실시간으로 phishing site 수집하여 24시간 주기로 업데이트 됨

Phishing Domain Database

Follow 1.2k

NOTICE: Do Not Clone the repository and rely on Pulling the latest info !!!

This WILL BREAK daily due to a complete reset of the repository history every 24 hours. Please rely **ONLY** on pulling individual list files or the full list of domains in tar.gz format and links in tar.gz format (updated hourly) using wget or curl.

Version: 22945 (2020-12-07 19:15:45 SAST)

Latest Threats @ 19:15:45	Active Threats Monday 2020-12-07	Total Links Discovered Today
347	526	697

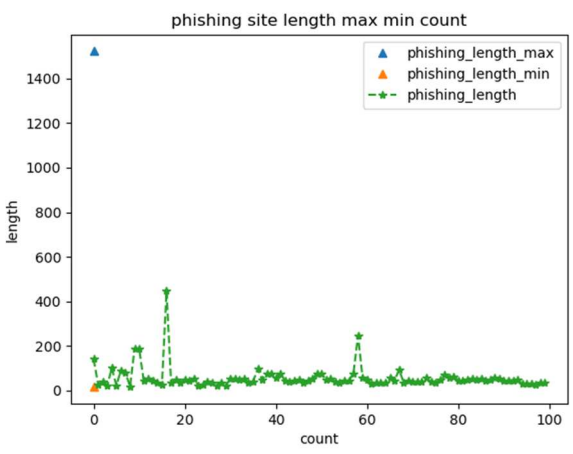
2-2). Phishing site feature 분석

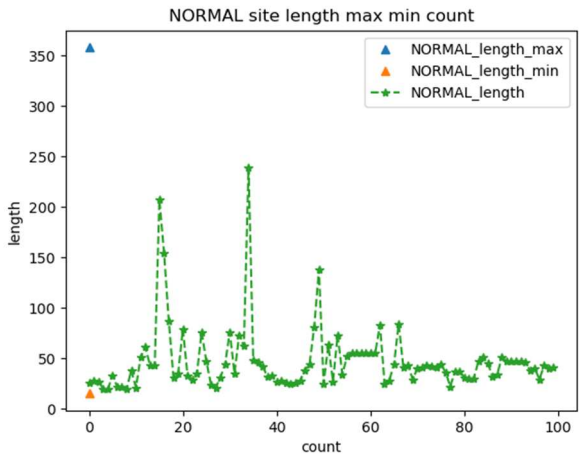
phishing site url 자체를 분석해봤다

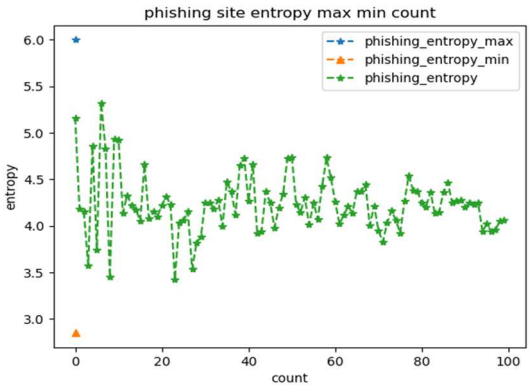
```
****** URL LENGTH AVERAGE ******
AVERAGE OF NORMAL URL LENGTH --> 56.11

Max NORMAL URL LENGTH --> 358
Min NORMAL URL LENGTH --> 15

****** URL ENTROPY AVERAGE ******
AVERAGE OF URL ENTROPY --> 4.30
NORMAL Max ENTROPY --> 5.89958509758102
NORMAL Min ENTROPY --> 2.895423870280316

PATH NORMAL AVERAGE OF URL ENTROPY --> 2.58
PATH NORMAL Max ENTROPY --> 5.542878556900888
PATH NORMAL Min ENTROPY --> 0.0
```

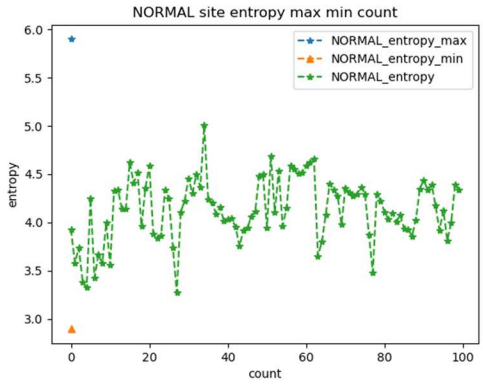
```
****** URL LENGTH AVERAGE ******
AVERAGE OF PHISHING URL LENGTH --> 88.92

Max PHISHING URL LENGTH --> 1524
Min PHISHING URL LENGTH --> 14

****** URL ENTROPY AVERAGE ******
AVERAGE OF URL ENTROPY --> 4.45
PHISHING Max ENTROPY --> 5.998872201885809
PHISHING Min ENTROPY --> 2.851018723094229

PATH PHISHING AVERAGE OF URL ENTROPY --> 3.57
PATH PHISHING Max ENTROPY --> 5.9438803071562045
PATH PHISHING Min ENTROPY --> -0.0
```

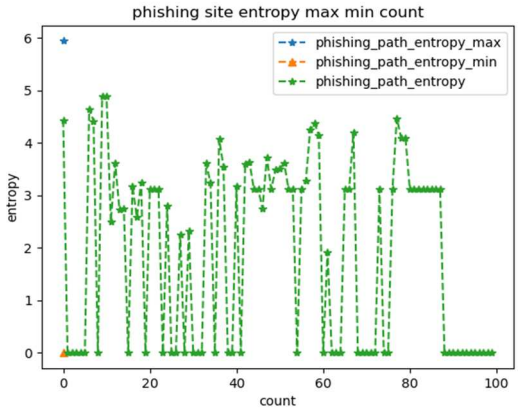
phishing site 가 normal site 보다 url길이가 길다는걸 확인 phishing길이 가 normal길이 보다 5배 정도 긴걸 확인 했다 전체 entropy 평균값은 phishing 과 normal 차이점이 없었지만 path 부분에 entropy는 phishing site 복잡도가 높은 걸 확인 하지만 phishing site중 bitly를이용한 url 축소를 사용한 경우가 있고 또한 최대값 과 최솟값의 표준편차가 매우 크기 때문에 의미가 크게 떨어졌다.

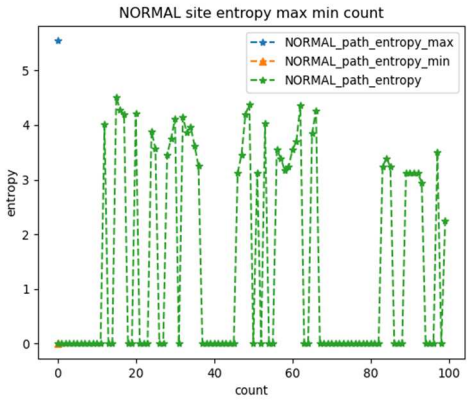








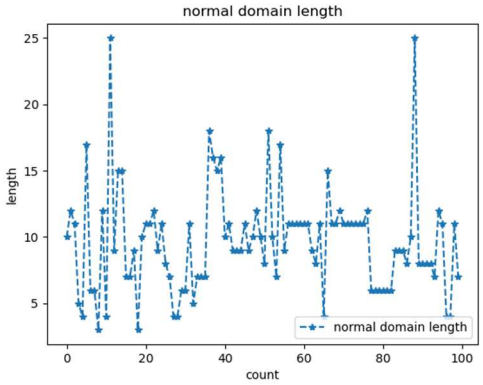


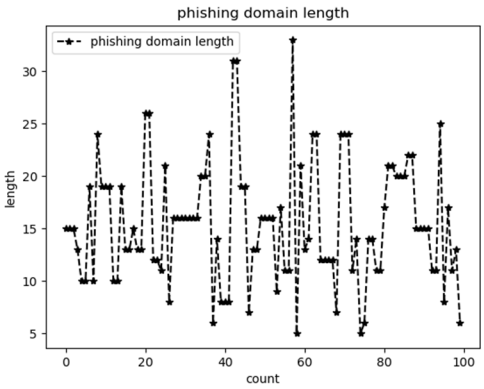


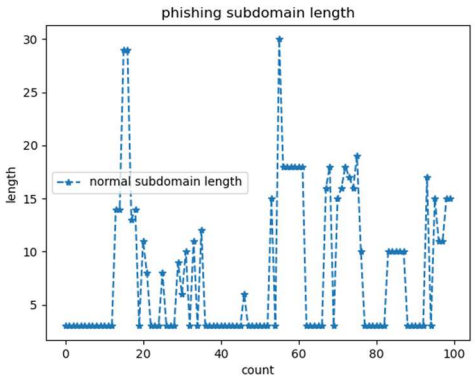

```
****** normal netloc ******
result NETLOC --> 60700 / not --> 694

****** normal suffix length ******
normal suffix max length --> 15 / min length --> 2
normal suffix average length --> 3.323956738443496

****** normal domain length ******
normal domain max length --> 63 / min length --> 1
normal domain average length --> 9.376470013356354

****** normal subdomain length ******
normal subdomain max length --> 55 / min length --> 0
normal subdomain average length --> 7.091947095807408

****** normal SpecialCharcter count ******
normal SpecialCharcter max count --> 23 / min length --> 1
normal SpecialCharcter average count --> 3.2822425644199757
```

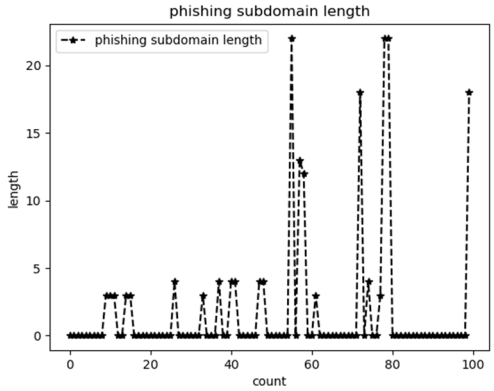
```
****** phishing netloc ******
result NETLOC --> 10324 / not --> 44776

****** phishing suffix length ******
phishing suffix max length --> 40 / min length --> 2
phishing suffix average length --> 3.5609009403017713

****** phishing domain length ******
phishing domain max length --> 61 / min length --> 1
phishing domain average length --> 11.718820613747358

****** phishing subdomain length ******
phishing subdomain max length --> 145 / min length --> 0
phishing subdomain average length --> 4.6016109045848825

****** phishing SpecialCharcter count ******
phishing SpecialCharcter max count --> 132 / min length --> 1
phishing SpecialCharcter average count --> 3.686896551724138
```

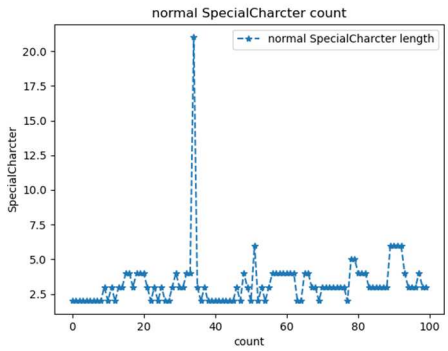
그외 domain, subdomain, suffix, 특수문자 전체적인 기준으로는 phishing site가 전체적으로 높았으나 표준편차는 양쪽 사이트 다 큰걸 알 수 있었다.

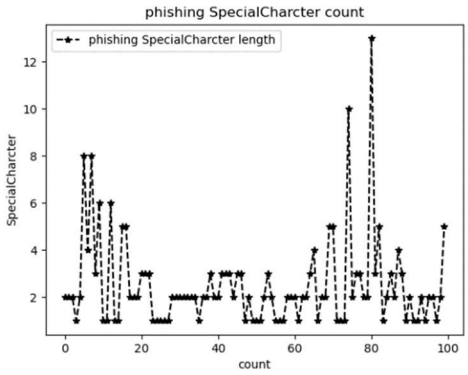

normal domain length
length
count
normal domain length

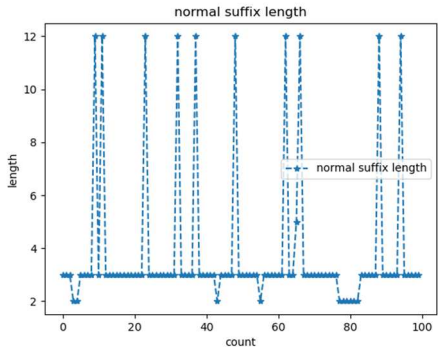

phishing domain length
length
count
phishing domain length

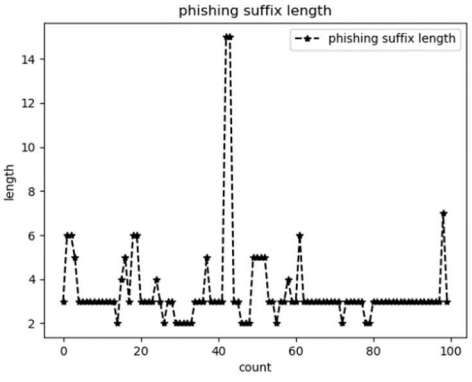

phishing subdomain length
length
count
normal subdomain length


phishing subdomain length
length
count
phishing subdomain length


normal SpecialCharcter count
SpecialCharcter
count
normal SpecialCharcter length


phishing SpecialCharcter count
SpecialCharcter
count
phishing SpecialCharcter length

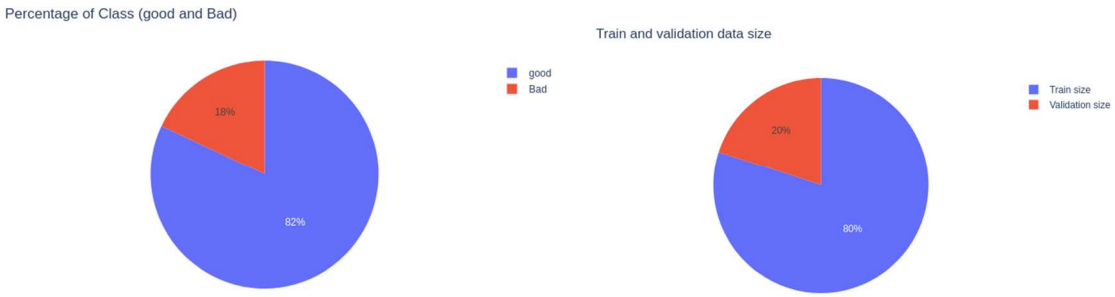

normal suffix length
length
count
normal suffix length


phishing suffix length
length
count
phishing suffix length

3) Phishing site feature 추출 및 ML(DL) 적용 결과 및 정탐률

1. Deep Learning

Kaggle dataset (42만개)로 진행했으며 데이터 분석한 결과 다음과 같은 결과 가 나왔다

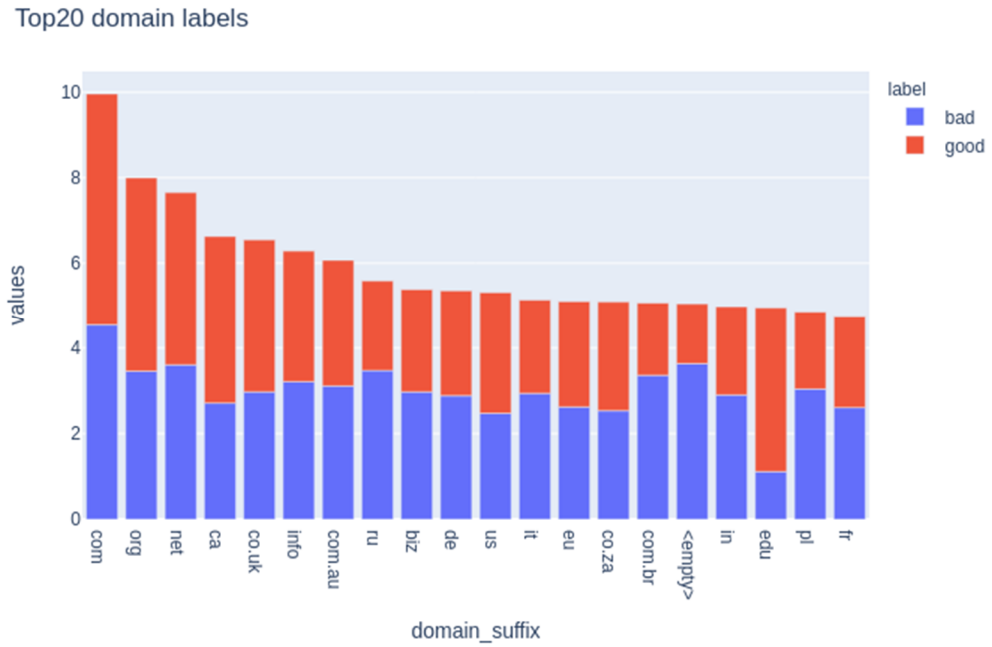


정상 URL 82%(34만) 비정상 URL(7만) 18% 확인이 됐으며

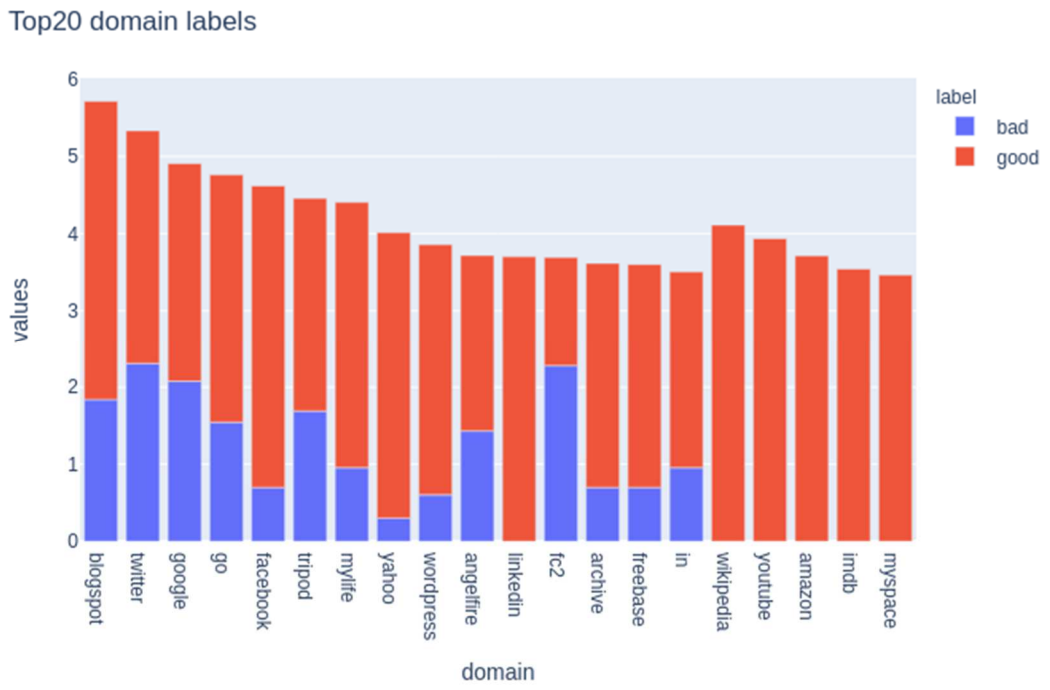
전체 데이터 비율을 8:2 비율을 산정하였고(validation size = 20%로 선정했다)

데이터 전처리 대상을 (url, label) → 칼럼 3 개 추가 (domain subdomain, domain_suffix)로 잡았으며 전처리 기준은 피싱사이트 특정적으로 URL 과 Subdomain 이 매우 길거나 아님 URL 자체가 매우 길고 나머지 정보가 없거나 생략되어 있는 경우를 확인했다

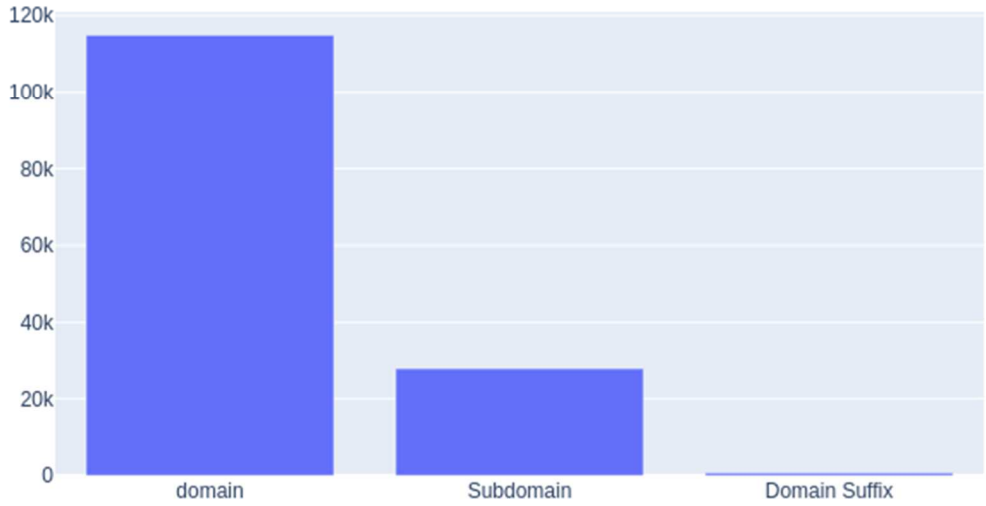
domain_suffix, Site 분석한 결과 다음과 같은 결과 가 나왔다



Domain_suffix TOP20 에 데이터 순위로서 <empty>(suffix 가 없는 부분)에 bad가 압도적인 걸 확인



그러므로 domain 전체적인 부분에 대한 데이터 분석을 한 결과 다음과 같은 결과 가 나왔다



처음 데이터셋 전처리 1단계로서 각각 URL 마다 subdomain domain domain_suffix로 나누었고 각 결측치 마다 <empty>로 채워 넣어 데이터 분석 및 전처리 효율을 높였음

	subdomain	domain	domain_suffix
0	<empty>	nobell	it
1	www	dghjdgf	com
2	<empty>	serviciosbys	com
3	mail	printakid	com
4	<empty>	thewhiskeydregs	com
...	...	...	...
1750349	<empty>	gerrydear	id.au
1750350	<empty>	opora-company	ru
1750351	<empty>	sportists	com
1750352	<empty>	hellohello-pension	com
1750353	<empty>	beautyevent	ru

위의 데이터 분석에 보았듯이 대부분 잘못된 레이블을 포함하는 경우가 많았지만 그렇지 않는 레이블도 존재했기에 CNN1D model input에 사용할 수 있도록 URL에서 토큰화를 진행했다

```
token_len -> 320
Before tokenization >
nobell.it/70ffb52d079109dca5664cce6f317373782/login.SkyPe
 .com/en/cgi-bin/verification/login/70ffb52d079109dca5664cce6f317373/index
 .php?cmd=_profile-ach&outdated_page_tmpl=p/gen/failed-to-load&nav=0.5.1&login_access=1322408526

After tokenization >
[12, 6, 29, 5, 17, 17, 4, 8, 3, 2, 33, 24, 34, 34, 29, 25, 20, 19, 24, 33, 31, 16, 24, 31, 19, 10, 7, 25, 30, 30, 27, 10, 10, 5,
 30, 34, 28, 16, 33, 28, 33, 28, 33, 32, 20, 2, 17, 6, 26, 8, 12, 4, 46, 36, 35, 55, 5, 4, 10, 6, 13, 2, 5, 12, 2, 10, 26, 8, 22,
  29, 8, 12, 2, 37, 5, 14, 8, 34, 8, 10, 7, 3, 8, 6, 12, 2, 17, 6, 26, 8, 12, 2, 33, 24, 34, 34, 29, 25, 20, 19, 24, 33, 31, 16,
 24, 31, 19, 10, 7, 25, 30, 30, 27, 10, 10, 5, 30, 34, 28, 16, 33, 28, 33, 28, 2, 8, 12, 19, 5, 40, 4, 11, 15, 11, 47, 10, 13,
 19, 42, 39, 11, 14, 6, 34, 8, 17, 5, 22, 7, 10, 15, 54, 6, 23, 3, 19, 7, 3, 5, 19, 39, 11, 7, 26, 5, 39, 3, 13, 11, 17, 42, 11,
 2, 26, 5, 12, 2, 34, 7, 8, 17, 5, 19, 22, 3, 6, 22, 17, 6, 7, 19, 54, 12, 7, 37, 42, 24, 4, 25, 4, 16, 54, 17, 6, 26, 8, 12, 39,
  7, 10, 10, 5, 9, 9, 42, 16, 28, 20, 20, 27, 24, 32, 25, 20, 30]
```

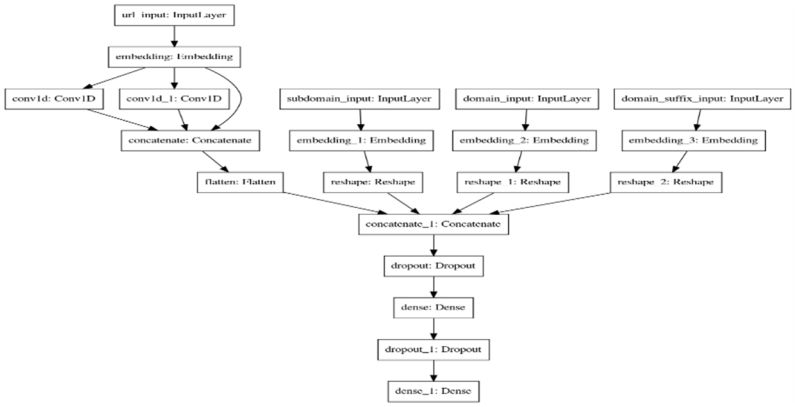
각각 URL 길이는 다르므로 균등하게 맞추기 위해서 패딩 작업을 실시했다

```
Before padding >
 [12, 6, 29, 5, 17, 17, 4, 8, 3, 2, 33, 24, 34, 34, 29, 25, 20, 19, 24, 33, 31, 16, 24, 31, 19, 10, 7, 25, 30, 30, 27, 10, 10, 5,
  30, 34, 28, 16, 33, 28, 33, 28, 33, 32, 20, 2, 17, 6, 26, 8, 12, 4, 46, 36, 35, 55, 5, 4, 10, 6, 13, 2, 5, 12, 2, 10, 26, 8,
 22, 29, 8, 12, 2, 37, 5, 14, 8, 34, 8, 10, 7, 3, 8, 6, 12, 2, 17, 6, 26, 8, 12, 2, 33, 24, 34, 34, 29, 25, 20, 19, 24, 33, 31,
 16, 24, 31, 19, 10, 7, 25, 30, 30, 27, 10, 10, 5, 30, 34, 28, 16, 33, 28, 33, 28, 2, 8, 12, 19, 5, 40, 4, 11, 15, 11, 47, 10,
 13, 19, 42, 39, 11, 14, 6, 34, 8, 17, 5, 22, 7, 10, 15, 54, 6, 23, 3, 19, 7, 3, 5, 19, 39, 11, 7, 26, 5, 39, 3, 13, 11, 17, 42,
 11, 2, 26, 5, 12, 2, 34, 7, 8, 17, 5, 19, 22, 3, 6, 22, 17, 6, 7, 19, 54, 12, 7, 37, 42, 24, 4, 25, 4, 16, 54, 17, 6, 26, 8, 12,
  39, 7, 10, 10, 5, 9, 9, 42, 16, 28, 20, 20, 27, 24, 32, 25, 20, 30]
After padding >
 [12  4 46 36 35 55  5  4 10  6 13  2  5 12  2 10 26  8 22 29  8 12  2 37
  5 14  8 34  8 10  7  3  8  6 12  2 17  6 26  8 12  2 33 24 34 34 29 25
 20 19 24 33 31 16 24 31 19 10  7 25 30 30 27 10 10  5 30 34 28 16 33 28
 33 28  2  8 12 19  5 40  4 11 15 11 47 10 13 19 42 39 11 14  6 34  8 17
  5 22  7 10 15 54  6 23  3 19  7  3  5 19 39 11  7 26  5 39  3 13 11 17
 42 11  2 26  5 12  2 34  7  8 17  5 19 22  3  6 22 17  6  7 19 54 12  7
 37 42 24  4 25  4 16 54 17  6 26  8 12 39  7 10 10  5  9  9 42 16 28 20
 20 27 24 32 25 20 30]
```

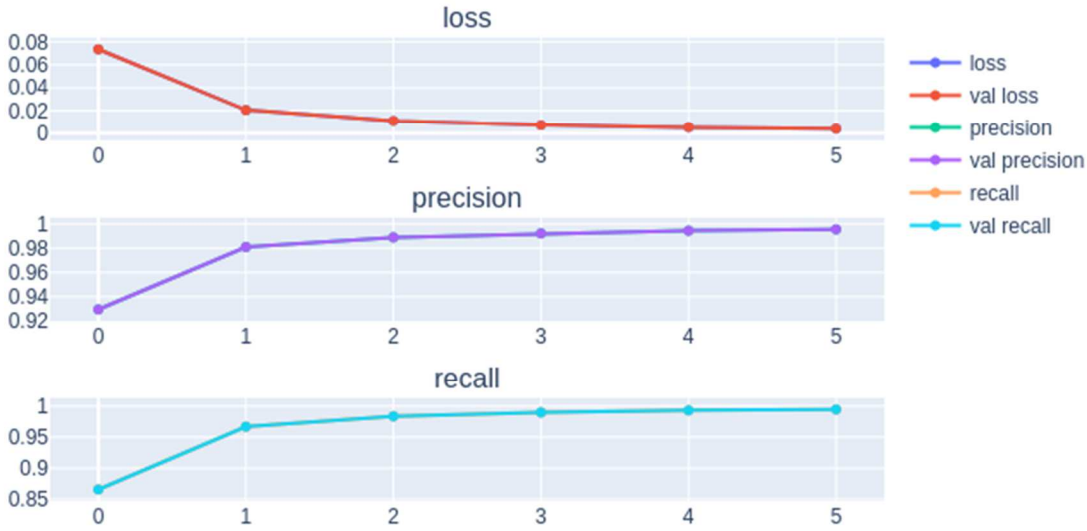
또한 subdomain, domain, domain_suffix 레이블을 정수화 하여 인코딩 해주었고

다음 labeling 값을 normal: 0, phishing: 1 값을 맞춘 다음 모델설계를 했으며 첫번째 모델은

Conv1D + Embedding



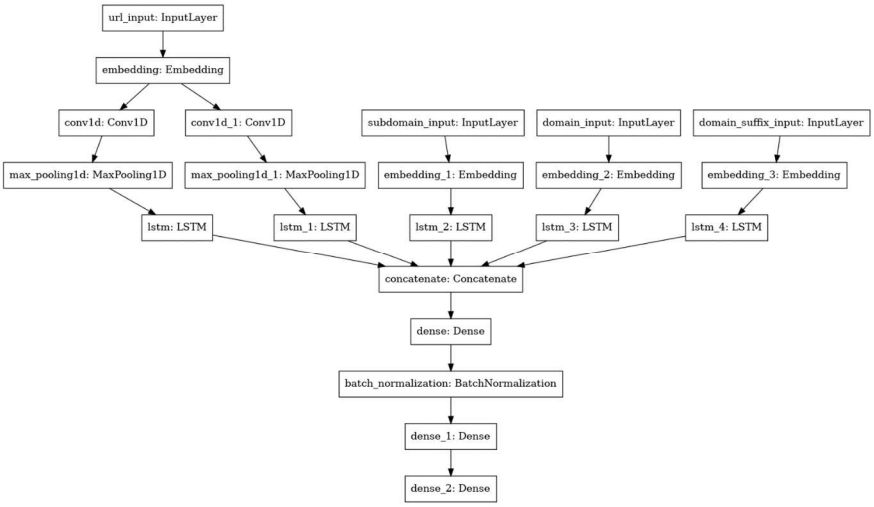
이 모델은 4개의 input 이 있고 첫번째 input은 토큰화 와 패딩을 수행한 URL으로 했고 기타 입력(domain, subdomain, domain_suffix)에서는 임베딩 레이어 넣었다 첫번째 input은 임베딩 레이어와 CONV1D 레이어를 통과하고 다음 입력은 임베딩 레이어만 통과하여 학습을 진행한 결과



acc: 98 / precision 98 recall 99 / loss: 0.5 이하의 값을 보여줬지만

phishing site validation 돌려본 결과 실질적으로 90% 성능을 보여줌으로써 많은 개선을 느낌

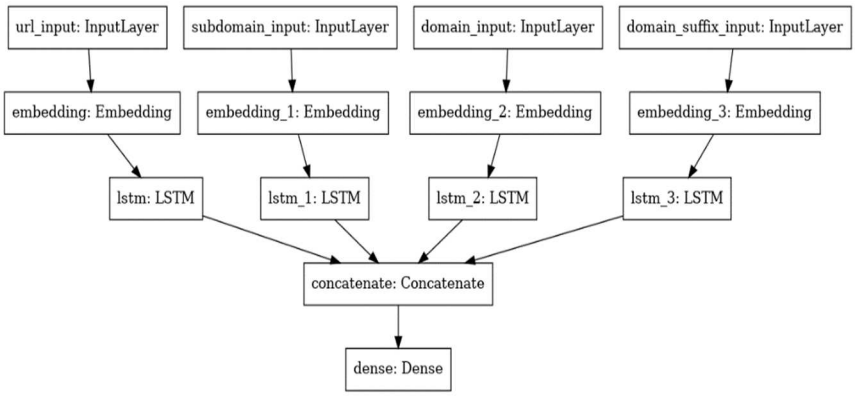
그 이후 CNN+LSTM, LSTM 다양하게 응용해서 돌려봤으나



Classification Report >

	precision	recall	f1-score	support
0	0.93	1.00	0.96	344821
1	0.99	0.67	0.80	75643
accuracy			0.94	420464
macro avg	0.96	0.84	0.88	420464
weighted avg	0.94	0.94	0.94	420464

CNN1D+ Embedding



Classification Report >

	precision	recall	f1-score	support
0	0.94	0.99	0.96	344821
1	0.95	0.69	0.80	75643
accuracy			0.94	420464
macro avg	0.95	0.84	0.88	420464
weighted avg	0.94	0.94	0.94	420464

LSTM

정밀도(precision) 부분에서 좋은 결과를 나타낼 수 있었으나 재현율(recall)부분에 약세가 보여 오탐률이 높았으며 사실상 모델 효력이 없다고 봐야했다.

Precision(정밀도) → 모델의 입장에서 Phishing 탐지 비율

Recall(재현율) → 실제 정답(data)의 입장에서 Phishing 탐지 비율

2. Machine Learning(tokenized)

데이터셋에 .exe, .virus 같은 단어가 데이터셋에 포함되어있는걸 확인하여 tokenizer를 사용하여 단어를 수집하고. URL을 벡터 형식으로 변환해보았다.

RegexpTokenizer

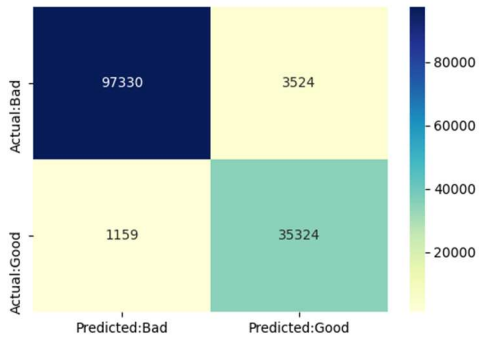
토큰 또는 토큰 간의 구분 기호와 일치하는 정규식을 사용하여 문자열을 분할하는 tokenizer다.

```
첫번째 url -->
nobell.it/70ffb52d079109dca5664cce6f317373782/login.SkyPe
 .com/en/cgi-bin/verification/login/70ffb52d079109dca5664cce6f317373/index
 .php?cmd=_profile-ach&outdated_page_tmpl=p/gen/failed-to-load&nav=0.5.1&login_access=1322408526
result -->
['nobell', 'it', 'ffb', 'd', 'dca', 'cce', 'f', 'login', 'SkyPe', 'com', 'en', 'cgi', 'bin', 'verification', 'login', 'ffb', 'd',
  'dca', 'cce', 'f', 'index', 'php', 'cmd', 'profile', 'ach', 'outdated', 'page', 'tmpl', 'p', 'gen', 'failed', 'to', 'load',
 'nav', 'login', 'access']
```

SnowballStemmer 작업으로 SnowballStemmer("English") 하여 형태소 분류 진행

```
stemmer of URL --->
0    [nobel, it, ffb, d, dca, cce, f, login, skype,...
1    [www, dghjdgf, com, paypal, co, uk, cycgi, bin...
2    [serviciosbi, com, paypal, cgi, bin, get, into...
3    [mail, printakid, com, www, onlin, americanexp...
4    [thewhiskeydreg, com, wp, content, theme, wide...
Name: stemmer, dtype: object
result stemmer --> 45.93476724624634
```

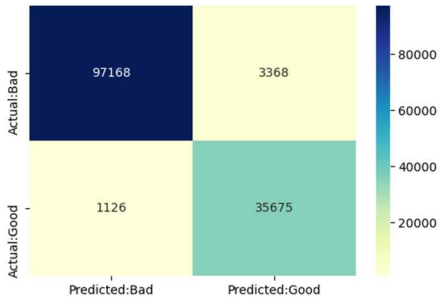
그다음 각 변화한 문자 별로 CountVectorizer(텍스트 말뭉치/토큰 수) 적용해서 벡터 변화 후 Logistic regression에 학습한 결과



CLASSIFICATION REPORT

	precision	recall	f1-score	support
bad	0.99	0.96	0.98	100781
good	0.91	0.97	0.94	36556
accuracy			0.97	137337
macro avg	0.95	0.97	0.96	137337
weighted avg	0.97	0.97	0.97	137337

Logistic regression 학습성과 가 좋아 pipline을만들어 Logistic + RegexpTokenizer 적용하여 튜닝후학습을 다시 돌려본 결과



CLASSIFICATION REPORT

	precision	recall	f1-score	support
bad	0.99	0.97	0.98	100536
good	0.91	0.97	0.94	36801
accuracy			0.97	137337
macro avg	0.95	0.97	0.96	137337
weighted avg	0.97	0.97	0.97	137337

다음과 같은 결과 가 나왔다 다만 이 모델에 학습 데이터의 오점은 http 에 대한 구분점이 전혀 안되어있어 http 가 포함된 URL 이 나올 때 오탐률이 매우 심해지는 결과가 나와 http 자체를 URL 스키마에 추가하여 시도해보면 또다시 다른 학습 결과가 나올 수 있다 .

Tokenized 데이터 도 학습 자체를 문자로 했기 때문에 https: http www 에 대한 학습이 진행되지 않아 오탐이 높아지는게 당연하여 학습한게 무의미했다

3. Domain Entropy (Machine Learning)

1 차 시도 로 42 만개의 데이터셋 시도를 진행해보았다

```
Head of data  --->
   Unnamed: 0                                url    label  result
0           0       https://www.google.com   benign       0
1           1      https://www.youtube.com   benign       0
2           2     https://www.facebook.com   benign       0
3           3        https://www.baidu.com   benign       0
4           4  https://www.wikipedia.org   benign       0
--------------------------------
Tail of data  --->
        Unnamed: 0  ... result
450171      450171  ...      1
450172      450172  ...      1
450173      450173  ...      1
450174      450174  ...      1
450175      450175  ...      1

[5 rows x 4 columns]
--------------------------------
Null check  --->
Unnamed: 0    0
url           0
label         0
result        0
dtype: int64
```

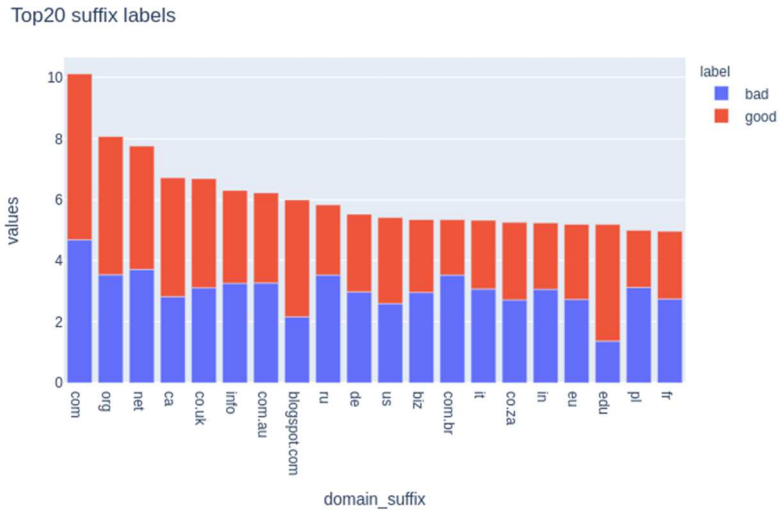
데이터 타입은 다음과 같이 이루어 있으며 쓸모없는 칼럼 Unnamed: 0 칼럼 제거

labeling 을 benign → good / malicious → bad 로 labeling 을 교체 해주었다(데이터 합병 할때 일반화 과정)

Phishing site 특징은 자신의 목적성을 숨겨 URL를 비이상적으로 길게 나타내는 경우가 있어 다음과 같은 3개의 칼럼을 추가하여 결과를 도출해본 결과

```
2st columns add Head of data  --->
                        url label  result  url_length  hostname_length  \
0     https://www.google.com  good       0          22               14
1    https://www.youtube.com  good       0          23               15
2   https://www.facebook.com  good       0          24               16
3      https://www.baidu.com  good       0          21               13
4  https://www.wikipedia.org  good       0          25               17

   path_length
0            0
1            0
2            0
3            0
4            0
------------------------------
2st columns add Tail of data  --->
                                                      url label  result  \
450171        http://ecct-it.com/docmmmnn/aptgd/index.php   bad       1
450172  http://faboleena.com/js/infortis/jquery/plugin...   bad       1
450173  http://faboleena.com/js/infortis/jquery/plugin...   bad       1
450174                            http://atualizapj.com/   bad       1
450175  http://writeassociate.com/test/Portal/inicio/I...   bad       1

        url_length  hostname_length  path_length
450171          43               11           25
450172         159               13          139
450173         147               13          127
450174          22               14            1
450175         143               18          118
------------------------------
```

Phishing site URL이 비이상적으로 확인할 수 있는 부분과 path길이가 상이한 걸 확인할 수 있다
Domain-suffix TOP20의 결과



suffix 칼럼을 추가할 시 com 의 일반화 가 매우 커지므로 오탐 발생 가능성이 생길 수 있어 suffix 칼럼을 삭제하였고 각각 [-@? %/=] http/https/www 카운터를 추가하여 경우의 수를 높이고

isnumeric(숫자체크(문자열 도 가능한 걸로 알고 있음),

isalpha(문자열 체크)을 적용하여 digit count 적용하여

```
data["count-digits"] = data["url"].apply(lambda digit: digit_count(digit))
data['count-letters']= data['url'].apply(lambda letter: letter_count(letter))
data['count_dir'] = data['url'].apply(lambda letter: no_of_dir(letter))
```

```
data["count-"] = data["url"].apply(lambda bar: bar.count("-"))
data['count@'] = data['url'].apply(lambda at: at.count('@'))
data['count?'] = data['url'].apply(lambda i: i.count('?'))
data["count%"] = data["url"].apply(lambda i: i.count("%"))
data["count."] = data["url"].apply(lambda point: point.count("."))
data["count="] = data["url"].apply(lambda equel: equel.count("="))
data["count-http"] = data["url"].apply(lambda http: http.count("http"))
data["count-https"] = data["url"].apply(lambda https: https.count("https"))
data["count-www"] = data["url"].apply(lambda www: www.count("www"))
```

```
3st Columns additional head of data  --->
                      url label  result  url_length  hostname_length  \
0     https://www.google.com  good       0          22               14
1   https://www.youtube.com  good       0          23               15
2  https://www.facebook.com  good       0          24               16
3     https://www.baidu.com  good       0          21               13
4 https://www.wikipedia.org  good       0          25               17

   path_length  fd_length  tld_length  count-  count@  count?  count%  count.  \
0            0          0          22       0       0       0       0       2
1            0          0          23       0       0       0       0       2
2            0          0          24       0       0       0       0       2
3            0          0          21       0       0       0       0       2
4            0          0          25       0       0       0       0       2

   count=  count-http  count-https  count-www  count-digits  count-letters  \
0       0           1            1          1             0             17
1       0           1            1          1             0             18
2       0           1            1          1             0             19
3       0           1            1          1             0             16
4       0           1            1          1             0             20

   count_dir
0          0
1          0
2          0
3          0
4          0
```

```
3st Columns additional Tail of data  --->
                                                    url label  result  \
450171        http://ecct-it.com/docmmmnn/aptgd/index.php   bad       1
450172  http://faboleena.com/js/infortis/jquery/plugin...   bad       1
450173  http://faboleena.com/js/infortis/jquery/plugin...   bad       1
450174                          http://atualizapj.com/      bad       1
450175  http://writeassociate.com/test/Portal/inicio/I...   bad       1

        url_length  hostname_length  path_length  fd_length  tld_length  \
450171          43               11           25          8          43
450172         159               13          139          2         159
450173         147               13          127          2         147
450174          22               14            1          0          22
450175         143               18          118          4         143

        count-  count@  count?  count%  count.  count=  count-http  \
450171       1       0       0       0       2       0           1
450172       0       0       0       0       2       1           1
450173       0       0       0       0       1       1           1
450174       0       0       0       0       1       0           1
450175       1       0       0       0       4       0           1

        count-https  count-www  count-digits  count-letters  count_dir
450171            0          0             0             34          3
450172            0          0            21            118         12
450173            0          0            20            109         12
450174            0          0             0             17          1
450175            0          1             9            118          7
```

정상사이트에 비해 phishing site 의 URL 길이의 세부적인 값이 높은 걸 확인할 수 있었음

도메인에 아이피 사용 여부를 판단하기 위해 정규표현식을 넣어 사용했고

```
def having_ip_address(url):
    match = re.search(
        '(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.'
        '([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\/)|'  # IPv4
        '((0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\/)' # IPv4 in hexadecimal
        '(?:[a-fA-F0-9]{1,4}:){7}[a-fA-F0-9]{1,4}', url)  # Ipv6
    if match:
        return -1
    else:
        return 1

def shortening_service(url):
    match = re.search('bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.im|is\.gd|cli\.gs|'
                      'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl\.nl|snipurl\.com|'
                      'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.com|fic\.kr|loopt\.us|'
                      'doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.ly|bit\.do|t\.co|lnkd\.in|'
                      'db\.tt|qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.ly|bit\.ly|ity\.im|'
                      'q\.gs|is\.gd|po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\.com|cutt\.us|u\.bb|yourls\.org|'
                      'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.net|1url\.com|tweez\.me|v\.gd|'
                      'tr\.im|link\.zip\.net',
                      url)
    if match:
        return -1
    else:
        return 1
```

다음과 같이 데이터 칼럼을 완성하였음

```
Index(['url', 'label', 'result', 'url_length', 'hostname_length',
       'path_length', 'fd_length', 'tld_length', 'count-', 'count@', 'count?',
       'count%', 'count.', 'count=', 'count-http', 'count-https', 'count-www',
       'count-digits', 'count-letters', 'count_dir', 'use_of_ip', 'short_url'],
      dtype='object')
```

본 데이터 칼럼 ML 에 적용하여 다음과 같은 값을 도출 해내였다

x 값 17 개 칼럼 y 값 result 칼럼 1 개 사용

- Decision Tree

```
Shape of x --> (450176, 17)
Shape of y --> (450176,)
**********************************************
Decision Accuracy Score > 0.9958555997004354
Decision Confusion_Matrix --->
[[241358    571]
 [   735  72460]]
```

- Random Forest

```
Forest Accuracy Score > 0.9971344613548953
Forest Confusion_Matrix --->
[[241642    287]
 [   616  72579]]
```

- Logistic regression

```
Logistic Accuracy Score > 0.9963728563993857
Logistic Confusion_Matrix --->
[[241399    530]
 [   613  72582]]
```

- ensemble LightGBM(하이퍼 파라미터 미적용)

```
ACC of lgbm --> 0.9971439814168391


classification report -->
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    241929
           1       1.00      0.99      0.99     73195

    accuracy                           1.00    315124
   macro avg       1.00      1.00      1.00    315124
weighted avg       1.00      1.00      1.00    315124
```

이 데이터 칼럼은 매우 단순하여 학습률은 높았으나 오탐이 존재하였고

특수문자(-$#@!) 칼럼을 하나로 합치고 각각 도메인 마다 쪼개 entropy 를 만들어 다시 데이터 칼럼을 만들었고

Head of data -->

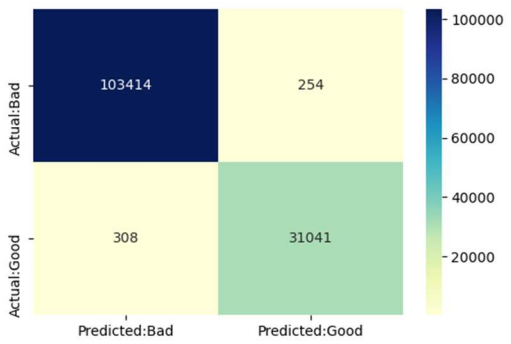
	url	url_length	entropy	pathentropy	domainentropy	tldentropy	subdomainentropy	FldEntropy	hostname_length	path_length	fd_length	tld_length	count-	count-@	special_chacter	count-http	count-https	count-www	count-digit	count-letter	count_dir	use_of_ip	result
0	https://www.google.com	22.0	3.663533	0.0	1.918296	1.584963	0.0	2.646439	14	0	0	3	1	1	2	1	1	1	0	17	0	1	0
1	https://www.youtube.com	23.0	3.762267	0.0	2.521641	1.584963	0.0	3.095795	15	0	0	3	1	1	2	1	1	1	0	18	0	1	0
2	https://www.facebook.com	24.0	3.855389	0.0	2.750000	1.584963	0.0	3.022055	16	0	0	3	1	1	2	1	1	1	0	19	0	1	0
3	https://www.baidu.com	21.0	3.880180	0.0	2.321928	1.584963	0.0	3.169925	13	0	0	3	1	1	2	1	1	1	0	16	0	1	0
4	https://www.wikipedia.org	25.0	3.813661	0.0	2.641604	1.584963	0.0	3.334679	17	0	0	3	1	1	2	1	1	1	0	20	0	1	0

Tail of data -->

	url	url_length	entropy	pathentropy	domainentropy	tldentropy	subdomainentropy	FldEntropy	hostname_length	path_length	fd_length	tld_length	count-	count-@	special_chacter	count-http	count-https	count-www	count-digit	count-letter	count_dir	use_of_ip	result
450049	http://ecct-it.com/docmmmnn/aptgd/index.php	43.0	3.874426	3.692879	2.235926	1.584963	0.0	2.845351	11	25	8	3	-1	1	2	1	0	0	0	34	3	1	1
450050	http://faboleena.com/js/infortis/jquery/plugin...	159.0	4.895229	4.891645	2.725481	1.584963	0.0	3.238901	13	139	2	3	1	1	3	1	0	0	21	118	12	1	1
450051	http://faboleena.com/js/infortis/jquery/plugin...	147.0	4.831358	4.792292	2.725481	1.584963	0.0	3.238901	13	127	2	3	1	1	2	1	0	0	20	109	12	1	1
450052	http://atualizapj.com/	22.0	3.720129	0.000000	2.846439	1.584963	0.0	3.467720	14	1	0	3	1	1	1	1	0	0	0	17	1	1	1
450053	http://writeassociate.com/test/Portal/inicio/I...	143.0	5.352063	5.410772	3.093069	1.584963	0.0	3.392147	18	118	4	3	-1	1	4	1	0	1	9	118	7	1	1

데이터 entropy 칼럼 중 전체적으로 phishing site entropy 가 높은걸 확인

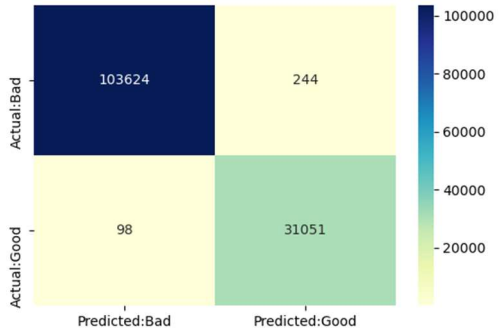
- Decision Tree



decision classification report -->

	precision	recall	f1-score	support
0	1.00	1.00	1.00	103722
1	0.99	0.99	0.99	31295
accuracy			1.00	135017
macro avg	0.99	0.99	0.99	135017
weighted avg	1.00	1.00	1.00	135017

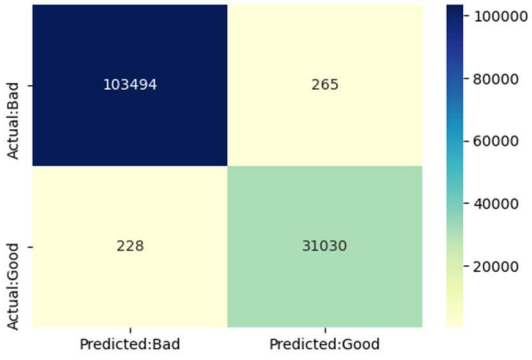
- Random Forest



forest classification report -->

	precision	recall	f1-score	support
0	1.00	1.00	1.00	103722
1	1.00	0.99	0.99	31295
accuracy			1.00	135017
macro avg	1.00	1.00	1.00	135017
weighted avg	1.00	1.00	1.00	135017

- Logistic Regression



logit classification report -->

	precision	recall	f1-score	support
0	1.00	1.00	1.00	103722
1	0.99	0.99	0.99	31295
accuracy			1.00	135017
macro avg	1.00	0.99	0.99	135017
weighted avg	1.00	1.00	1.00	135017

3 자 데이터 정탐률 [6:4(오탐)] ~ [7:3(오탐)]


```
{
    "Timestamp": "2020-11-17 23:14:17.812169",
    "URL": "url sample",
    "detection": true,
    "module": "ML_PhishingDetected",
    "log": [
        {
            "submodule": 0,
            "internal_url": "http://3.131.17.188/wordpress",
            "external_url": "https://www.cosmosfarm.com/products/kboat
            "result": "0",
            "percentage": "99%"
        },
        {
            "submodule": 0,
            "internal_url": "http://3.131.17.188/wordpress",
            "external_url": "https://wordpress.org",
            "result": "1",
            "percentage": "98%"
        },
        {
            "submodule": 0,
            "internal_url": "http://3.131.17.188/xe",
            "external_url": "http://xpressengine.github.io/XEIcon",
            "result": "1",
            "percentage": "100%"
        },
        {
            "submodule": 0,
            "internal_url": "http://3.131.17.188/xe",
            "external_url": "https://github.com/xpressengine/XEIcon",
            "result": "1",
            "percentage": "89%"
        },
```

```
{
    "submodule": 0,
    "internal_url": "http://3.131.17.188/xe",
    "external_url": "https://www.xpressengine.com/download",
    "result": "0",
    "percentage": "100%"
},
{
    "submodule": 0,
    "internal_url": "http://3.131.17.188/xe",
    "external_url": "https://github.com/xpressengine/xe-core",
    "result": "1",
    "percentage": "85%"
},
{
    "submodule": 0,
    "internal_url": "http://3.131.17.188/xe",
    "external_url": "https://www.xehub.io",
    "result": "0",
    "percentage": "99%"
},
{
    "submodule": 0,
    "internal_url": "http://3.131.17.188/xe",
    "external_url": "https://www.xpressengine.com",
    "result": "0",
    "percentage": "98%"
}
```

```
{
    "submodule": 0,
    "internal_url": "http://3.131.17.188/xe",
    "external_url": "https://www.xpressengine.com/forum",
    "result": "0",
    "percentage": "99%"
},
{
    "submodule": 0,
    "internal_url": "http://3.131.17.188/xe",
    "external_url": "https://www.xpressengine.com/qna",
    "result": "0",
    "percentage": "99%"
},
{
```

4. DomainEntropy + Requset

데이터를 제구성하여 23295 개의 데이터로 진행했으며

Entropy 칼럼에 Request 칼럼을 더해 7 개 칼럼을 더하였다

1. DomainRegisterationLength

Phishing site 는 짧은 시간 동안 활동하고 또한 신뢰할 수 있는 도메인은 정기적으로 지급(갱신)되는 것으로 보고 있기에 도메인 등록 기간을 feature 점을 넣었다 지금 가지고 있는 데이터의 phishing site 는 최대 600 일 정도이기에 다음과같이 feature 점을 넣었다

```
if 도메인 기간 <= 600 일:
        phishing site
else:
        normal site
```

2. Google index

이 특징은 웹사이트가 google index 에 있는지 검사한다 google 에 의해 indexing 되면 검색 결과에 표시되는데 일반적으로 phishing site 는 단기간동안 활동하므로 phishing site 는 google index 에 찾을 수 없다.

```
if 웹 페이지가 google index 존재:
        normal site
else:
        phishing site
```

3. Iframe

Iframe 은 현재 표시된 추가 웹페이지를 표시하는데 사용되는 HTML 태그인데 phishing site 는 “iframe”태그를 사용하여 프레임 경계 없이 보이지 않게 할 수 있다

```
if <iframe> tag 사용:
        phishing site
else:
        normal site
```

4. Server Form Handler (SFH)

Empty string 또는 “about: blank”을 포함하는 SFH 는 제출된 정보(데이터)의해 조치를 취해야 하기 때문에 의심스러운 것으로 여겨진다. 또한 SFH 의 domain 이 웹페이지 domain 과 다를 경우, 외부 도메인에 의해 거의 처리되지 않기 때문에 웹페이지가 의심스럽다는 것을 보여줌.

```
if SFH 가 “about:blank” 이거나 비어있음:
    phishing site
elif SFH 처리 도메인이 다른 도메인으로 되어있음:
    의심되는 사이트
Else:
    normal site
```

5. Favicon

Favicon 은 특정 웹페이지와 관련된 그래픽 이미지(아이콘)이다.

웹페이지의 사이트이름 옆에 표시되는 아이콘이 현재 도메인 이외 외부 도메인에서 Favicon 이 이루어진 경우 Phishing site 으로 간주할 수 있다

Color Scripter

```
if 외부 도메인으로 load 된 Favicon:
        phishing site
else:
        normal site
```

6. Submitting email

웹 양식은 사용자가 자신의 data 를 제출하고 처리하기 위해 서버로 보내지는데 phishing site 는 개인 이메일로 사용자의 정보를 redirect 할 수 있다 이 과정을 위해 php mail() 함수

Html mailto 태그 속성을 사용하여 redirect 를 할 수 있다.

```
 if mail() 함수 또는 mailto 함수를 사용해서 정보를 보내는가:
        phishing site
else:
        normal site
```

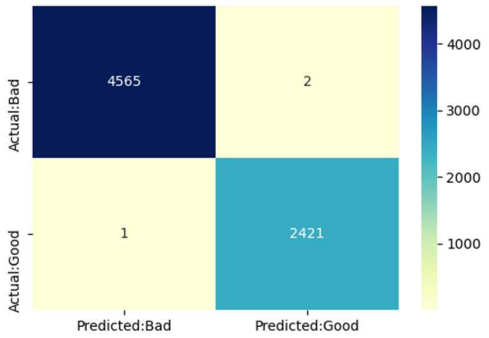
7. Web traffic

사람들이 얼마나 방문했는지 alexa 데이터 베이스에서 웹페이시 순위를 볼 수 있는데 phishing site 의 경우 짧은 기간만 활동하기 때문에 alexa 에 확인되지 않을 가능성이 높다

```
if alexa 랭킹 <= 100,000 위에 드는가?:
        normal site
elif alexa 랭킹 > 100,000 위에 들지 않는가?:
        의심 사이트
else:
        phishing site
```

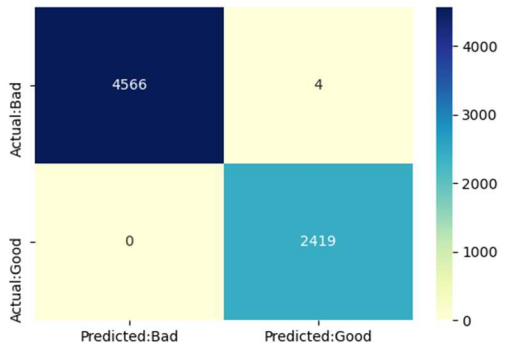
학습 결과

- Decision Tree



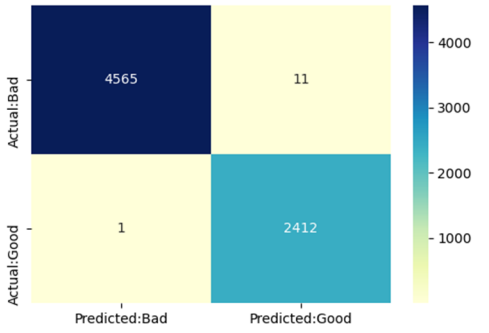

```
decision classification report -->
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4566
           1       1.00      1.00      1.00      2423

    accuracy                           1.00      6989
   macro avg       1.00      1.00      1.00      6989
weighted avg       1.00      1.00      1.00      6989
```

- Random forest




```
forest classification report -->
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4566
           1       1.00      1.00      1.00      2423

    accuracy                           1.00      6989
   macro avg       1.00      1.00      1.00      6989
weighted avg       1.00      1.00      1.00      6989
```

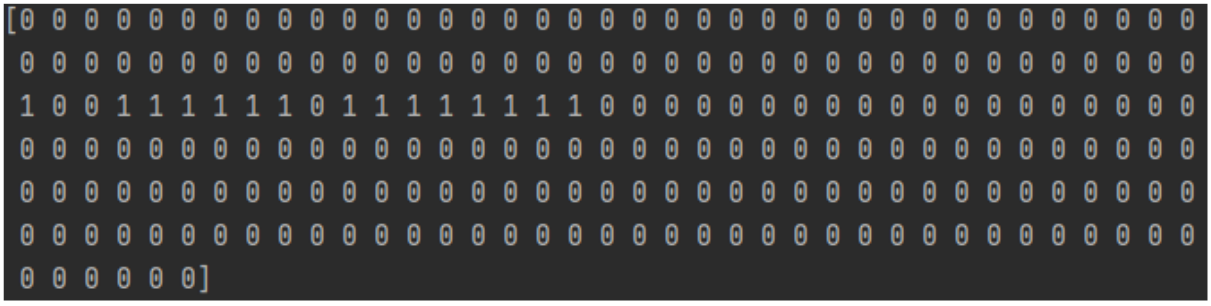
- Logistic Regression




```
logit classification report -->
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4566
           1       1.00      1.00      1.00      2423

    accuracy                           1.00      6989
   macro avg       1.00      1.00      1.00      6989
weighted avg       1.00      1.00      1.00      6989
```

3 자 데이터 정탐률 (100 개) 기준

85 개 정탐 15 개 오탐

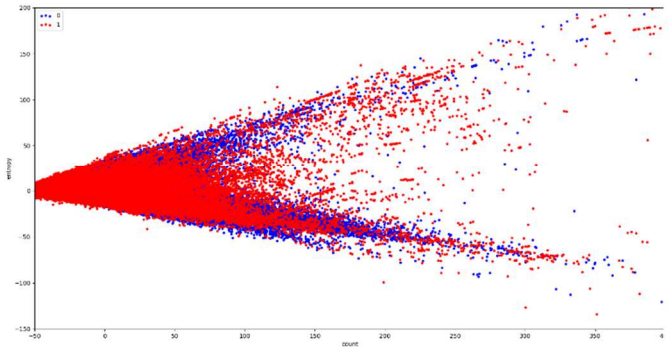


Unsupervised Learning(PCA 주성분 분석)

Supervised learning 에서 방향을 틀어 각 phishing site 마다 feature 점을 뽑아 cluster 을 확인해보았다.

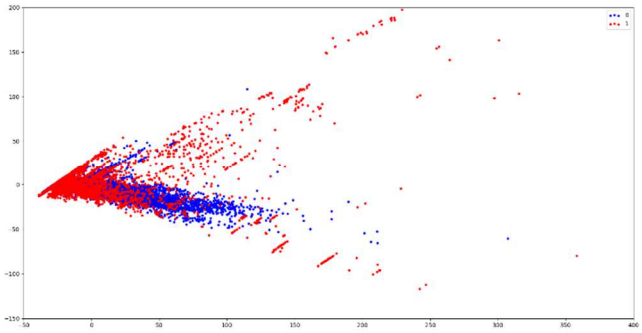
1. Entropy feature 만 뽑았을 때

red: phishing, blue: normal

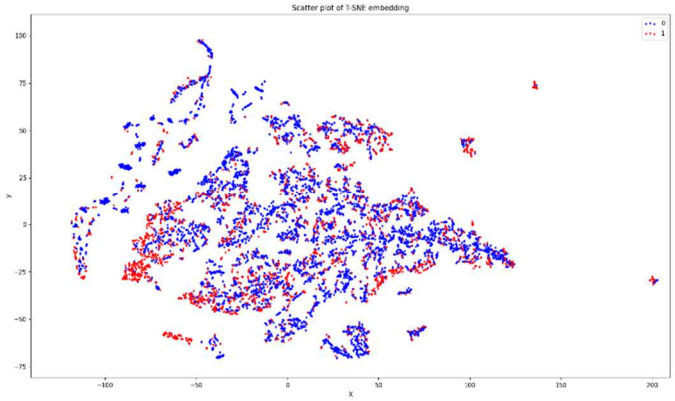


2. Entropy request feature 추출

red: phishing, blue: normal



T-SNE (Entropy PCA training)



T-SNE(Entropy request training)