



Artificial Intelligence & Security

Ji-hyun Lee
(jerry_jane@naver.com)

Introduction

What is Artificial intelligence ?

- Artificial Intelligence (AI) refers to the capability of a computer system
- to perform tasks that typically require human intelligence.
- These tasks include learning, reasoning, problem-solving, perception,
- language understanding, and even speech recognition.

Overview

- 1. The Implementation of AI
 - Strengthening Cyber Security Company
- 2. [Practice] Training/Using a Spam Email Classifier
- 3. Issues with Generative AI Models & Latest Solutions
 -

The Implementation of AI Strengthening Cyber Security Company



Barracuda **Sentinel**

AI for Real-Time Spear Phishing
and Cyber Fraud Defense

Barracuda Sentinel focuses on email security, utilizing AI to identify and thwart spear-phishing attempts and account takeover attacks.
(AI로 이메일/피싱/계정탈취 공격 식별-방지)



CYLANCE

Cylance PROTECT is a pioneer in using AI for threat prevention, leveraging machine learning to predict and prevent both known and unknown malware threats.
(머신러닝으로 알려지지 않은 위협까지 탐지/대응)



CROWDSTRIKE

(AI로 END Point
위협 탐지/방지)



Vectra AI specializes in network threat detection, utilizing AI to analyze network behaviors and swiftly identify and mitigate cyber threats.
(네트워크 위협 탐지/분석)

Source : <https://youtu.be/GqVGS-y8YnY?si=mgU3Nok4A0pARyKz> (Simplilearn)

실습1. 스팸메일 분류기

머신러닝 기반의 간단한 스팸메일
분류기를 학습하여 만들어봅시다.

1. 데이터
전처리

2. 모델학습

3. 분류모델
사용

- [Practice] Training/Using a Spam Email Classifier

(1) 데이터 전처리

- 필요한 라이브러리와 데이터를 불러옵니다.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

df = pd.read_csv(r'D:\240122\mail_data.csv')
```

(1) 데이터 전처리

2. 불러온 데이터를 살펴봅니다.

df.head()		
	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

INDEX : 행 구분,
0부터 시작

Category : 비/스팸 구분

Message : 메일 텍스트

(1) 데이터 전처리

2. 불러온 데이터를 살펴봅니다.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Category    5572 non-null   object  
 1   Message     5572 non-null   object  
dtypes: object(2)
memory usage: 87.2+ KB
```

총 행 : 5572행

데이터 타입

Category : object (문자)

Message : object (문자)

(1) 데이터 전처리

3. 만약 결측치*가 있다면 결측치를 처리한다.

```
data = df.where(pd.notna(df), '')
```

위 구문은 NaN이 아닌 요소는 그대로 두고,
NaN인 값을 빈 문자열 ("")로 대체하는 작업을 수행한다.

Nan 값을 그대로 두게 될 경우,
머신러닝이 학습하는 시점에서 문제가 발생할 수 있음.

*결측치
:이상치(Outlier)의
하위개념,
데이터에 값이 없음을
의미

Non/NONE/nan 값
등으로 표기.

(1) 데이터 전처리 - 원-핫인코딩

4. 원-핫 인코딩*을 통해 컴퓨터가 처리하기 쉽게 만들어줍니다.

```
# 'Category' 열이 'spam'인 행의 'Category' 값을 0으로 설정  
data.loc[data['Category'] == 'spam', 'Category'] = 0
```

```
# 'Category' 열이 'ham'인 행의 'Category' 값을 1으로 설정  
data.loc[data['Category'] == 'ham', 'Category'] = 1
```

Spam은 0으로, 정상 메일은 1로 표기합니다.

*원-핫인코딩
◦ 카테고리별 이진 특성을 만들어 해당하는 0,1 값으로 처리해주는 것

(2) 모델학습

1. 모델을 학습시키기 위해, 각각의 컬럼들을 X와 Y로 나누어 줍니다.

- ```
X = data['Message']
Y = data['Category']
#각 변수(X, Y)에 컬럼 값을 할당합니다.
```

## (2) 모델학습

1. 모델을 학습시키기 위해, 각 각의 컬럼들을 X와 Y로 나누어 줍니다.

| X                                                      |
|--------------------------------------------------------|
| 0 Go until jurong point, crazy.. Available only ...    |
| 1 Ok lar... Joking wif u oni...                        |
| 2 Free entry in 2 a wkly comp to win FA Cup fina...    |
| 3 U dun say so early hor... U c already then say...    |
| 4 Nah I don't think he goes to usf, he lives aro...    |
| ...                                                    |
| 5567 This is the 2nd time we have tried 2 contact u... |
| 5568 Will ü b going to esplanade fr home?              |
| 5569 Pity, * was in mood for that. So...any other s... |
| 5570 The guy did some bitching but I acted like i'd... |
| 5571 Rofl. Its true to its name                        |
| Name: Message, Length: 5572, dtype: object             |

| Y                                           |
|---------------------------------------------|
| 0 1                                         |
| 1 1                                         |
| 2 0                                         |
| 3 1                                         |
| 4 1                                         |
| ..                                          |
| 5567 0                                      |
| 5568 1                                      |
| 5569 1                                      |
| 5570 1                                      |
| 5571 1                                      |
| Name: Category, Length: 5572, dtype: object |

## (2) 모델학습

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state =3)
```



### X, Y

X와 Y를 각각 'Train' 과 'Test' 세트로 나눕니다.

### Test\_size = 0.2

이 때, 'Train' 과 'Test'는 각각 8:2 비율입니다. (Test 비율이 항상 적습니다.)

Random\_state = 3 난수화 지정 번호입니다. 이 번호를 지정할 경우,

동일한 결과를 가지게 됩니다.

## (2) 모델학습

2. 텍스트 데이터를 수치형 데이터로 변환하여  
자연어 처리 (NLP)나 머신러닝 모델에 적용할 수 있도록 합니다.

```
feature_extraction = TfidfVectorizer(min_df =1, stop_words = 'english', lowercase ='True')
#불용어로 영어를 사용합니다. 또한 소문자로 변환합니다
```

TfidfVectorizer

참고)

텍스트 데이터를 TF-IDF (Term Frequency-Inverse Document Frequency) 값으로  
변환하는 데 사용되는 scikit-learn 라이브러리의 클래스

TF(Term Frequency): 특정 단어가 문서 내에서 얼마나 자주 나타나는지

IDF(Inverse Document Frequency): 해당 단어가 전체 문서 집합에서 얼마나 희귀한지

TF-IDF 값은 특정 단어가 특정 문서에서 얼마나 중요한지를 측정합니다.

## (2) 모델학습 - 불용어란?

```
feature_extraction = TfidfVectorizer(min_df = 1, stop_words = 'english', lowercase = 'True')
#불용어로 영어를 사용합니다. 또한 소문자로 변환합니다
```

### 불용어(Stop Words)란

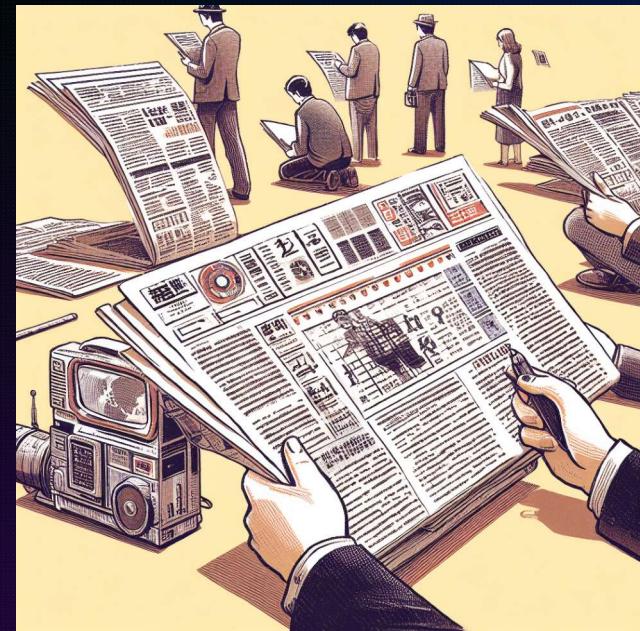
자연어 처리에서 흔히 사용되지 않거나  
덜 중요한 단어들을 말합니다.  
효율을 위해 제거해줍니다.

예를 들어, 어떤 글에서

"하늘은 푸르고, 강은 맑습니다."

(The sky is blue, and the river is clear.) "

라고 한다면, 여기서 덜 중요한 단어는 은, 고, 습니다. 일 것입니다.  
영어에서는 is와 and, the에 해당합니다. 불용어들입니다.



이미지 출처 : MicroSoft AI

## (2) 모델학습

```
X_train_features = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)
```

3. 훈련 데이터(X\_train), 테스트 데이터(X\_test)에 대해서  
특성 추출기(TfidfVectorizer)를 사용하여 TF-IDF 행렬을 생성합니다.

```
Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')
```

4. Y의 object를 0과 1로 변환했으니,  
Integer(정수)형으로 변환합니다.

```
Y_train.info()
<class 'pandas.core.series.Series'>
Int64Index: 4457 entries, 3075 to 1688
Series name: Category
Non-Null Count Dtype

4457 non-null int32
dtypes: int32(1)
memory usage: 52.2 KB
```

## (2) 모델학습

\*희소 행렬의 형태로 출력되어 크기와 그 안에 들어있는 값들이 표시됩니다.

```
print(X_train_features)
```

```
(0, 5413) 0.6198254967574347
(0, 4456) 0.4168658090846482
(0, 2224) 0.413103377943378
(0, 3811) 0.34780165336891333
(0, 2329) 0.38783870336935383
(1, 4080) 0.18880584110891163
(1, 3185) 0.29694482957694585
(1, 3325) 0.31610586766078863
(1, 2957) 0.3398297002864083
(1, 2746) 0.3398297002864083
(1, 918) 0.22871581159877646
(1, 1839) 0.2784903590561455
(1, 2758) 0.3226407885943799
(1, 2956) 0.33036995955537024
(1, 1991) 0.33036995955537024
(1, 3046) 0.2503712792613518
```

```
print(X_test_features)
```

```
(0, 7271) 0.1940327008179069
(0, 6920) 0.20571591693537986
(0, 5373) 0.2365698724638063
(0, 5213) 0.1988547357502182
(0, 4386) 0.18353336340308998
(0, 1549) 0.2646498848307188
(0, 1405) 0.3176863938914351
(0, 1361) 0.25132445289897426
(0, 1082) 0.2451068436245027
(0, 1041) 0.28016206931555726
(0, 405) 0.2381316303003606
(0, 306) 0.23975986557206702
(0, 20) 0.30668032384591537
(0, 14) 0.26797874471323896
(0, 9) 0.2852706805264544
(0, 1) 0.2381316303003606
```

\*희소 행렬이란?

성긴 행렬(sparse matrix)  
이라고도 하며,

행렬의 값이 대부분 0인  
경우를 가리키는 표현

(출처 : 위키백과)

## (2) 모델학습

5. 로지스틱 회귀 모델을 생성합니다.

```
model = LogisticRegression()
```



6. 모델을 학습(fit) 시킵니다.

```
model.fit(X_train_features, Y_train)
```

```
LogisticRegression()
```



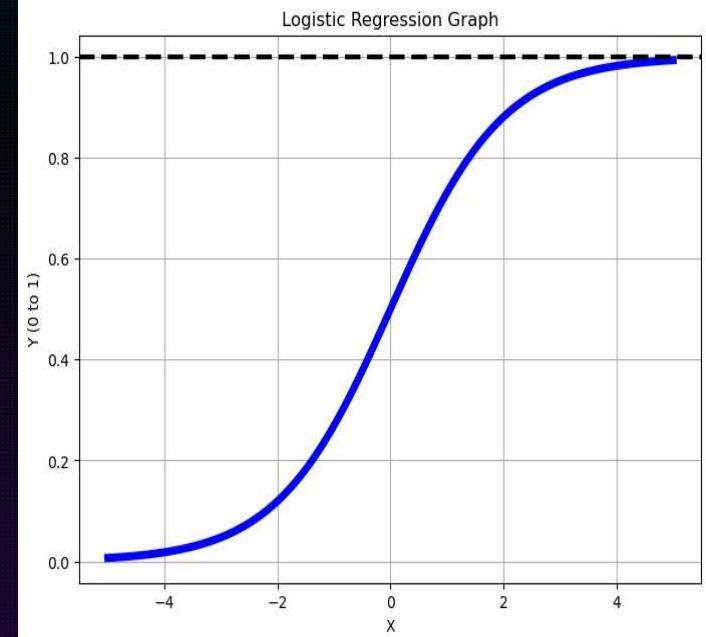
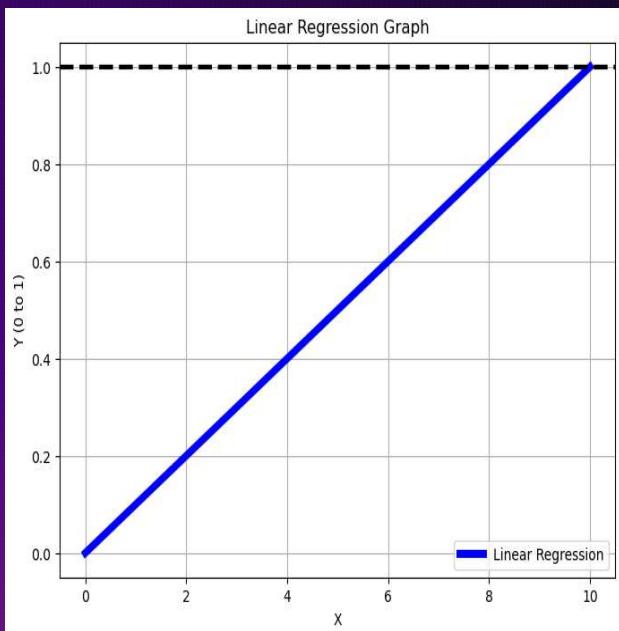
fit: 모델을 학습하는 메소드

학습 데이터와 타깃 변수를 인자로 받아 모델의 가중치를 조절

# 로지스틱 회귀(Logistic Regression) 란?

종속변수가 범주형인 경우에 적용하는 회귀분석.

종속변수  $Y$ 를 직접 모델링하지 않고,  $Y$ 가 특정 범주에 속할 확률을 구한다.



그래프를 통해 보는 선형회귀(Linear Regression)와의 차이

출처 : 파이썬 한권으로 끝내기 ADP 실기 (시대에듀)  
데싸라면, 빨간색 물고기, 자투리코드 저

## (2) 모델학습

7-1. 학습시킨 데이터를 바탕으로 Training 예측 값을 생성합니다.

```
prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)
```

| Name             | Description                     |
|------------------|---------------------------------|
| Predict()        | 모델을 사용하여 입력 데이터에 대한 예측 값을 생성한다. |
| Accuracy_score() | 예측값과 실제 레이블을 비교하여 정확도를 계산한다.    |

## (2) 모델학습

7-2. 학습시킨 데이터를 바탕으로 Test 예측 값을 생성합니다.

```
prediction_on_test_data = model.predict(X_test_features)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)
```

| Name             | Description                     |
|------------------|---------------------------------|
| Predict()        | 모델을 사용하여 입력 데이터에 대한 예측 값을 생성한다. |
| Accuracy_score() | 예측값과 실제 레이블을 비교하여 정확도를 계산한다.    |

## (2) 모델학습

- 

```
print('Accuracy on Training data', accuracy_on_training_data)
```

```
Accuracy on Training data 0.9670181736594121
```

- Training 데이터에 대한 정확도는 97%로 확인됩니다.

-

## (2) 모델학습

- ```
print('Accuracy on Test Data', accuracy_on_test_data)
```

Accuracy on Test Data 0.9659192825112107
- Test 데이터에 대한 정확도 또한 97%로 확인됩니다.

(3) 분류 모델 사용

모델이 생성되었으니, 사용해보겠습니다.

```
input_your_mail = ["My Dearest. the inspiring story of people who find hope and positivity even in the  
input_data_features = feature_extraction.transform(input_your_mail)  
  
prediction = model.predict(input_data_features)
```

“My Dearest. the inspiring story of people....” 얼마 전 TV에서 방영된。
드라마 연인에 대한 위키백과 소개글을 갖다 주었습니다.

(3) 분류 모델 사용

```
input_your_mail = ["My Dearest. the inspiring story of people who find hope and positivity even in the  
input_data_features = feature_extraction.transform(input_your_mail)  
  
prediction = model.predict(input_data_features)
```

```
print(prediction)  
  
if(prediction[0]==1):  
    print("Normal mail")  
  
else:  
    print('Spam Mail')
```

-
이대로라면
0과 1로만 출력하여 헷갈릴 수 있으니
○
1일 경우 'Normal mail'로
0일 경우 'Spam Mail'로 표시하도록 작성합니다.
결과는?????----->

(3) 분류 모델 사용

```
input_your_mail = ["My Dearest. the inspiring story of people who find  
input_data_features = feature_extraction.transform(input_your_mail)  
  
prediction = model.predict(input_data_features)  
  
print(prediction)  
  
if(prediction[0]==1):  
    print("Normal mail")  
  
else:  
    print('Spam Mail')  
.  
[1]  
. Normal mail
```

○

드라마 연인에 대한 글은 스팸 메일이 아닙니다!

(3) 분류 모델 사용

사미 You
영어 스팸메일 예시 하나만 작성해줘

ChatGPT
물론이에요. 아래는 일반적인 영어 스팸 메일의 예시입니다. 스팸 메일은 일반적으로 광고, 사기, 혹은 부적절한 콘텐츠를 포함하고 있을 수 있습니다. 이는 특정한 실제 스팸 메일과는 무관하며, 예시를 위한 가상의 텍스트입니다.

```
vbnet
Subject: Exclusive Offer: Win a Luxury Vacation Now!
Dear [Your Name],
Congratulations! You have been selected as the lucky winner of our excl
To claim your prize, simply click on the link below and provide your pe
```

GPT에게 스팸 메시지 하나만
만들어 달라고 한 후,
그 메시지를 넣어봅니다.

(3) 분류 모델 사용

```
input_your_mail = ["Congratulations! You have been selected as the lucky winner of our exclusive promot  
input_data_features = feature_extraction.transform(input_your_mail)  
  
prediction = model.predict(input_data_features)  
  
print(prediction)  
  
if(prediction[0]==1):  
    print("Normal mail")  
  
else:  
    print('Spam Mail')  
  
[0]  
Spam Mail
```

스팸입니다.

Congratulations!!

The Newest AI and Security

3. Issues with Generative AI Models & Latest Solutions

AI 이미지 생성도구 'Stable Diffusion'이 무단으로 이미지를 추출/사용하여 Getty Images가 고소를 했습니다.

*이미지는 해당 AI가 Getty Image를 사용하여 만든 결과물

ARTIFICIAL INTELLIGENCE / TECH / LAW

Getty Images is suing the creators of AI art tool Stable Diffusion for scraping its content



An image created by Stable Diffusion showing a recreation of Getty Images' watermark. Image: The Verge / Stable Diffusion

/ Getty Images claims Stability AI 'unlawfully' scraped millions of images from its site. It's a significant escalation in the developing legal battles between generative AI firms and content creators.

By [James Vincent](#), a senior reporter who has covered AI, robotics, and more for eight years at The Verge.

Jan 17, 2023, 7:30 PM GMT+9 | □ 18 Comments / 18 New



아티스트들, 생산형 AI에 대한 저작권 침해 소송 1심 패배

GOT A TIP?   NEWSLETTERS [SUBSCRIBE](#)

NEWS FILM TV AWARDS LIFESTYLE BUSINESS GLOBAL VIDEO MUSIC LABOR LISTS

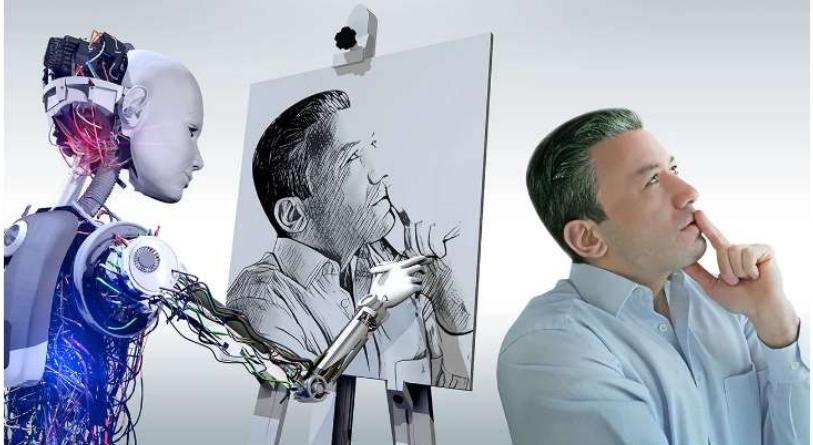
THR, ESQ.

HOME > BUSINESS > BUSINESS NEWS

Artists Lose First Round of Copyright Infringement Case Against AI Art Generators

While a federal judge advanced an infringement claim against Stability AI, he dismissed the rest of the lawsuit.

BY WINSTON CHO  OCTOBER 30, 2023 4:57PM



AD  ADVERTISEMENT
Donkeys Need Us
PETA They're desperately waiting for a kind person to help protect them.

WEEKLY NEWSLETTER
Unique expertise on how the law impacts Hollywood pros, projects

출처 :
TheHollywood
Reporter

일러스트레이터 'A' 모씨의 고민



내가 공들여 그린 그림을
AI로 학습시켜서
내 그림체를 복제하면 어쩌지?
그림을 안 올릴 수도 없고...

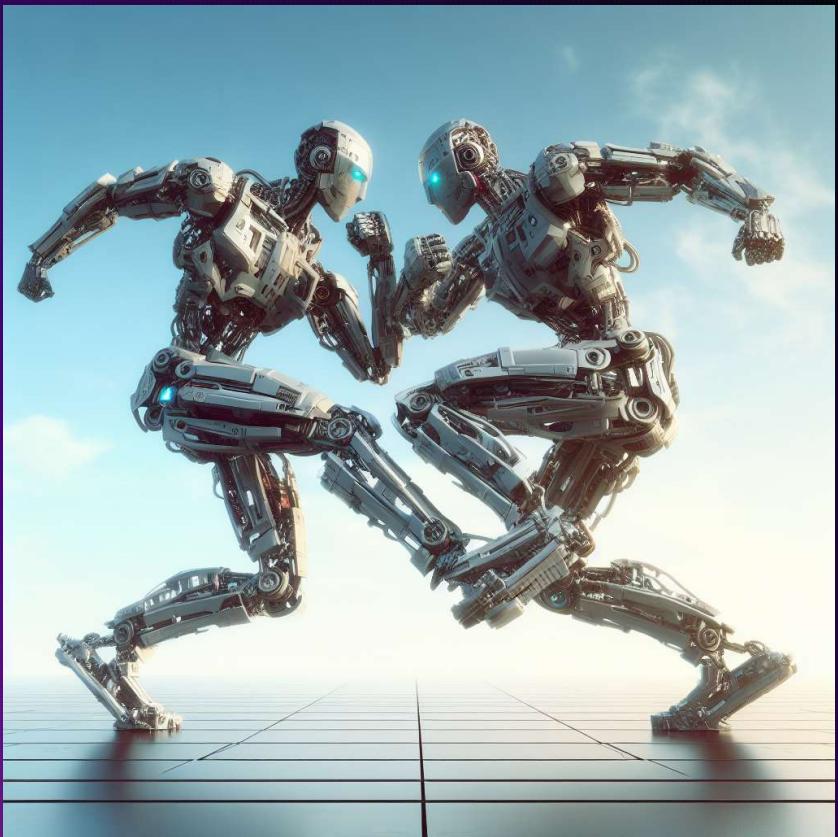
이미지 출처 : MicroSoft AI

이런 상황에서 . . .

AI가 방어할 수
있는 방법은
'정말'
없습니까?



이미지 출처 : MicroSoft AI



YES, WE CAN.

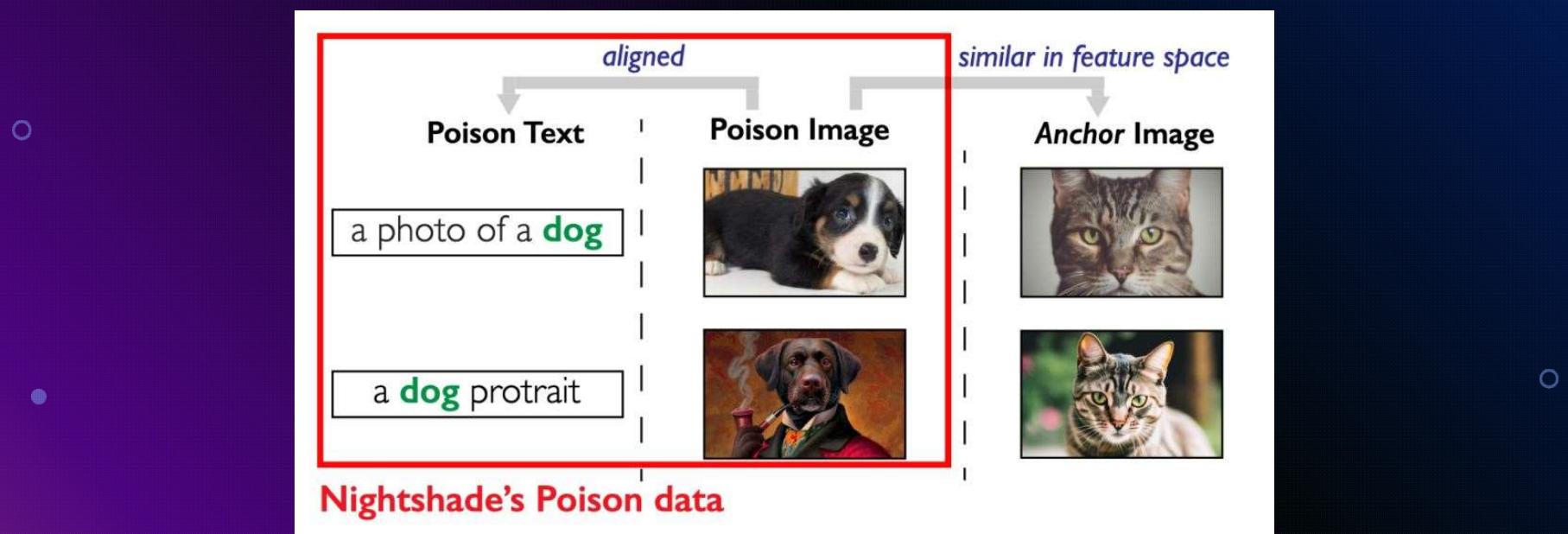
AI vs AI

NightShade

이미지 출처 : MicroSoft AI

NightShade

시카고 대학의 Ben Zhao 연구팀이 만든 AI 모델



생성형 AI가 저작권자의 허락 없이 학습에 이용할 수 없도록 혼돈을 주는 것
(독을 먹인다고 표현: Poisoning)

NightShade

생성형 AI에게 “Cubism(입체파)”를 만들어 달라고 했을 경우



독을 먹지 않은 모델



NightShade에 의해
독을 '100'번 먹은 모델



NightShade에 의해
독을 '300'번 먹은 모델

Cubism이 아닌 Anime 형태가 됨

이미지출처 : MIT Technology Review :
This new data poisoning tool lets artists fight back against generative AI

NightShade

생성형 AI에게 “HAT (모자)”를 만들어 달라고 했을 경우



독을 먹지 않은 모델



NightShade에 의해
독을 '100'번 먹은 모델



NightShade에 의해
독을 '300'번 먹은 모델

HAT이 아닌 CAKE 형태가 됨

이미지출처 : MIT Technology Review :
This new data poisoning tool lets artists fight back against generative AI

NightShade

작동 예시)



NightShade 진행



일종의 인공지능을 공격하는 것
*Adversarial Attack (적대적 공격) 유사

1. 창작자는 자신의 그림을
업로드하기 전 NightShade를
통해 Poisoning 시킨다.

2. 독을 먹인 이미지는
육안으로는 동일하게 보이나
학습할 AI에게는 다르게 인식됨

이미지 출처 : MicroSoft AI

[결과]
허가 받지 않은 작품들을
학습시켜, 비슷한 그림체를
구현하고자 했으나 Poison으로 인해
의도치 않은 전혀 다른 작품이 생성.

Adversarial Attack (적대적 공격)

'노이즈'를 추가하여 기계학습 모델을 속여서 잘못된 분류나 예측을 유발하는 공격입니다.

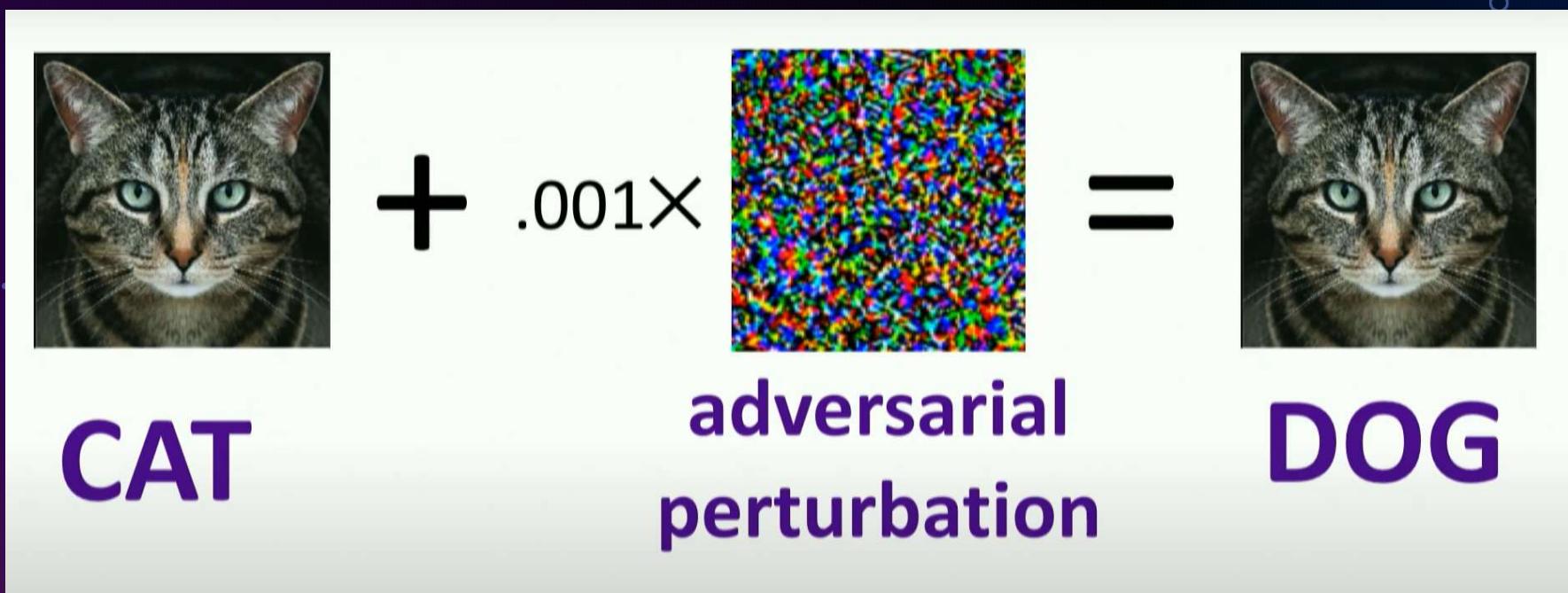


사진 출처 : I.J. Goodfellow, J.Shlens and C.Szegedy. Explaining and harnessing adversarial examples. 2015

기존 공격방식인 'Adversarial Attack'을 저작권자를 보호하는 도구로 사용

-

NightShade



생성형 AI

시사점)
공격 도구는 방어 도구가
될 수 있으며,
앞으로 이런 이슈들은
계속 발생할 것.
이를 막는 것 또한 AI로
가능하다는 점.

이미지 출처 : MicroSoft AI

Reference & Source

1. AI 적용한 사이버 기업들

Simplilearn Youtube Channel

<https://youtu.be/GqVGS-y8YnY?si=mgU3Nok4A0pARyKz>

2. 머신러닝을 이용한 스팸 분류기

Simplilearn Youtube Channel

<https://www.youtube.com/watch?v=FkF2jhaRJIs>

3. 발표자료 중 로지스틱 회귀 부분

파이썬 한권으로 끝내기 : ADP 실기 대비 , 시대에듀
-데싸라면, 빨간색물고기, 자투리 코드 저

4. 자문/인용

CHAT-GPT

5.1.Nightshade/ MIT Magazine/

<https://www.technologyreview.com/2023/10/23/1082189/data-poisoning-artists-fight-generative-ai/>

5.2. Adversarial Attack 부분

Two Minute Papers Youtube Channel

<https://www.youtube.com/watch?v=AOZw1tgD8dA>

5.3. RSA Conference Youtube Channel

https://www.youtube.com/watch?v=3hig_oEz8Rg

5.4. NightShade 부분

Gamefromscratch Youtube Channe

<https://www.youtube.com/watch?v=MGEtHqsFnHU>

5.5. The verge

<https://www.theverge.com/2023/1/17/23558516/ai-art-copyright-stable-diffusion-getty-images-lawsuit>

5.6. TheHollywoodReporter

<https://www.hollywoodreporter.com/business/business-news/artists-copyright-infringement-case-ai-art-creators-1235632929/>

Thank you ^_^



이미지 출처 : MicroSoft AI

이지현
jerry.jane@naver.com

43