

멀티모달 기반 스팸 필터링 플랫폼 개발



201824534 윤상호

201824636 이강우

202055651 조재홍

지도교수 최윤희

목 차

| | |
|--|----|
| 1. 서론..... | 1 |
| 1.1. 연구 배경..... | 1 |
| 1.2. 연구 목표..... | 2 |
| 2. 연구 배경..... | 2 |
| 2.1. 멀티모달(Multi-Modal)기반 스팸 필터링 모델 개발..... | 2 |
| 2.2. 데이터 수집 및 전처리..... | 3 |
| 2.2.1. 데이터 수집..... | 3 |
| 2.2.2. 데이터 전처리..... | 4 |
| 2.3. 피처 분석..... | 6 |
| 2.4. 모델 구현 및 학습..... | 9 |
| 2.4.1. 모델 선정..... | 9 |
| 2.4.2. MLP 모델 구현 및 학습..... | 10 |
| 2.4.3. MLP 하이퍼 파라미터 튜닝..... | 13 |
| 2.5. 모델 성능 평가..... | 14 |
| 2.6. GUI 시각화 도구(WEB) 개발..... | 15 |
| 2.6.1. Client-Side..... | 15 |
| 2.6.2. Server-Side..... | 20 |
| 2.7. 스팸 필터링 모델과 웹 연동..... | 22 |
| 2.7.1. Polling..... | 22 |
| 2.7.2. Chart.js를 이용한 SPAM 판독과 시각화..... | 23 |
| 3. 연구결과 분석 및 평가..... | 25 |
| 4. 결론 및 향후 연구 방향..... | 27 |
| 5. 구성원별 역할 및 개발 일정..... | 28 |
| 6. 참고 문헌..... | 29 |

1. 서론

1.1. 연구 배경

전자우편(e-mail)은 영리 목적의 광고성 정보인 스팸 메일(spam-mail)과 정상적인 내용을 포함하는 햄 메일(ham-mail)로 구분할 수 있다. 스팸메일이란 전자우편이나 휴대전화 등 정보 통신서비스를 이용하는 이용자의 단말기로 본인이 원하지 않음에도 일방적으로 이용자에게 전송되는 영리 목적의 광고성 정보를 말한다. 그러나 최근에는 일반적으로 스팸메일을 상업적인 목적의 전자우편만으로 한정하지 않고, 본인이 원하지 않는 전자우편 또한 스팸메일이라고 부르기도 한다.

스팸메일의 큰 특징 중 하나는 대량성(bulk)으로, 프로그램을 통해 매우 손쉽게 불특정 다수에게 무수히 많은 양의 스팸메일을 반복적으로 전송할 수 있다는 점이다. 이러한 특징 때문에 스팸메일을 받는 이용자의 불편을 유발하고, 필요한 정보의 수신을 방해하는 등의 문제점을 야기할 수 있다.

스팸메일을 분류하는 기법으로는 규칙 기반(Rule-based)과 기계학습 기반(Machine-Learning-based)으로 나뉘어진다. 규칙 기반 스팸메일 필터링은 사용자가 정의한 문자열 집합에 따라서 스팸메일을 탐지한다. 하지만 Spammer 들에 의해 스팸메일의 내용이 필터링을 우회할 수 있도록 변하기 때문에 키워드나 규칙을 이용하는 것으로는 새로운 유형의 스팸메일 탐지의 정확도가 떨어진다.

기계학습 기반 스팸메일 필터링은 컴퓨터가 자동으로 메일의 특징을 판단하고 선정하여 필터링 규칙을 자동으로 생성할 수 있다. 이는 시간이 지남에 따라 새로운 유형의 스팸메일 대처에 유리하다는 장점을 가진다. 하지만, 기존에 존재하는 스팸메일 필터링의 경우 단일 데이터 형태, 즉 하나의 모달리티만을 학습하여 task 를 수행하기 때문에 가끔 이질적인 결과를 출력하는 한계점을 보인다. 따라서 본 졸업과제에서는 이 한계점을 극복하기 위해 다양한 모달리티를 학습에 이용하여 더욱 사람처럼 학습하고 결과를 추론하는 방법을 멀티모달 기반 스팸 필터링이라고 부르며 이를 활용하여 멀티모달을 활용한 텍스트 기반 스팸 필터링 모델 플랫폼을 개발하고자 한다.

1.2. 연구 목표

본 졸업과제는 멀티모달 딥러닝/머신러닝 기반 스팸 필터링 플랫폼 개발을 목표로 한다.

- 수집 텍스트 스팸 데이터 셋(kaggle), 생성 텍스트 스팸 데이터 셋(chat-gpt), 이미지 스팸 데이터 셋(Spam Archive, Flickr)의 3 가지 모달리티를 활용한다.
- 위 모달리티들을 기반으로 각 모달리티만의 특징을 분석하여 이를 기반으로 모델을 학습시키고 학습된 모델로 스팸메일 여부를 판별한다.
- 스팸 필터링 모델과 DB, 웹을 연동하여 시각화 인터페이스(스팸 메일 여부, 스팸 분류 이유 등 유의미한 정보)를 구현한다.
- 새로운 유형과 다양한 방식(이미지와 텍스트가 같이 오는 경우 등)의 스팸 메일에 대처에 유연한 스팸 필터링 모델을 개발한다.

2. 연구 배경

2.1. 멀티모달(Multi-Modal)기반 스팸 필터링 모델 개발

멀티모달 AI 는 텍스트, 이미지, 영상, 음성 등 다양한 데이터 모달리티를 함께 고려하여 서로의 관계성을 학습 및 표현하는 기술이다. 따라서 멀티모달 AI 는 하나의 모달리티를 활용하는 것보다 다양한 작업을 수행할 수 있다.

최근 전송되는 메일들은 단일 유형으로 구성된 메일 뿐만 아니라, 다양한 유형의 데이터로 구성된 메일들(이미지와 텍스트가 함께 동봉된 경우 등)이 상당수 존재한다. 기존의 스팸메일 필터링 모델은 주로 텍스트 유형의 메일만을 규칙 기반 혹은 기계학습 기반으로 필터링하기 때문에 다양한 유형으로 구성되어 있는 스팸메일에 대처하지 못한다는 특징이 있다.

이를 해결하기 위해 수집 텍스트 메일 데이터, 생성 텍스트 메일 데이터, 수집 이미지 메일 데이터를 모달리티로 사용하여 텍스트 기반의 특징들을 분석하고 추출하여

이미지와 텍스트가 동시에 메일로 전송되는 상황에서의 스팸메일 탐지가 가능한 모델, 즉 멀티모달을 활용한 스팸 필터링 모델을 개발하였다.

2.2. 데이터 수집 및 전처리

2.2.1. 데이터 수집

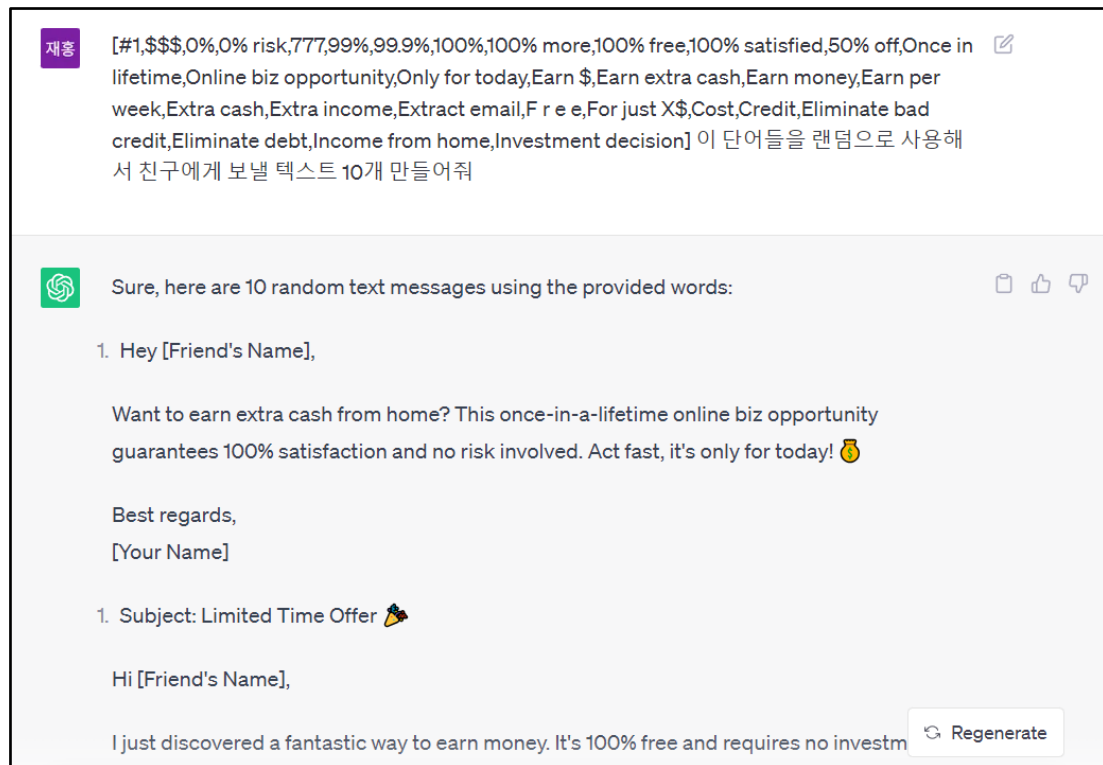
<수집 텍스트 메일 데이터>

Spammer들은 시간이 지남에 따라 기존의 스팸 필터링 모델을 우회하기 위해 다양한 형태로 스팸메일을 변화시킨다. 이러한 Concept Drift 현상을 고려하여 다양한 시점의 텍스트 스팸 데이터 셋을 Kaggle 공개 데이터 저장소에서 수집하였으며, 중복되는 데이터 셋은 모두 제외하고 최종적으로 6000여 개의 데이터를 확보하였다.

기존에는 약 6만 개의 이메일 데이터를 가진 Enron E-mail Dataset을 모델 학습에 사용하는 것을 고려하였으나, 생성 텍스트 데이터나 수집 이미지 데이터와의 극심한 데이터 수의 차이로 인해 최종 모델에서의 데이터 불균형 현상을 방지하고자 Kaggle 공개 데이터 저장소에서 수집한 6000여 개의 데이터를 모델 학습 및 성능 검증에 사용하기로 최종적으로 결정하였다.

<생성 텍스트 메일 데이터>

생성형 AI 서비스인 Chat GPT, Bard, Bing 등으로 텍스트 스팸 데이터를 생성을 시도하면, 불법적인 행위에 서비스를 제공하지 않는다면 서비스를 거부하는 제약사항이 존재한다. 이러한 제약사항을 우회하기 위해 다양한 유형의 스팸 키워드들을 부여하고 해당 키워드를 랜덤하게 선택해서 텍스트를 생성하라고 입력하는 방법을 사용하여 스팸메일을 생성하였다. 최종적으로 1200여 개의 생성 텍스트 메일 데이터를 확보하여 모델 학습 및 성능 검증에 사용하였다.



[그림 1] Chat GPT를 사용하여 생성한 스팸메일 예시

<수집 이미지 메일 데이터>

텍스트 기반의 스팸 필터링 모델이 목표이기 때문에 이미지에서 텍스트를 추출하여 학습에 사용한다. 따라서 수집하는 이미지 메일 데이터에는 텍스트가 반드시 포함되어 있어야 하는 제약사항이 존재한다. 이러한 제약사항을 우회하기 위해 스팸 이미지는 Spam Archive에서 텍스트를 포함한 이미지만 분류하여 수집하였다. Spam Archive에 있는 정상 이미지는 대부분이 이미지 내에 텍스트가 존재하지 않아서 Flickr 사진 및 비디오 공유 플랫폼에서 직접 수집하였다. 최종적으로 1200여 개의 수집 이미지 메일 데이터를 확보하여 모델 학습 및 성능 검증에 사용하였다.

2.2.2. 데이터 전처리

본 졸업과제에서는 사용하는 분류기가 총 5가지가 있다. TF-IDF 단일 피처 모델 3개, TF-IDF와 여러 분석한 피처를 추가한 다중 피처 모델 2개로 분류된다.

<TF-IDF>

메일 내용 그 자체를 입력으로 사용하고 학습 모델이 알아서 결과에 영향을 미치는 중요한 feature들을 판단하도록 TF-IDF(Term Frequency-Inverse Document Frequency, 이하 TF-IDF) Vectorize 기법을 사용하였다. TF-IDF 방식은 단어의 빈도와 역 문서 빈도(문서의 빈도에 특정 식을 취하는 것)를 사용하여 메일 내용 내의 토큰화 된 단어마다 중요한 정도에 따라서 가중치를 생성하여 모델 학습에 사용하는 것이다. 위에서 언급한 다중 입력 모델 또한 TF-IDF 방식에 분석한 feature들을 추가하여 학습을 진행하였다.

<토큰화>

토큰화는 TF-IDF 방식을 사용하기 위해 필요한 전처리 방법이다. 본 졸업과제에서는 영단어를 기준으로 단어 토큰화를 진행하였으며, TF-IDF에서 토큰화된 단어를 인코딩하여 숫자 값을 mapping 시킨 후, 가중치를 부여하였다. 토큰화는 본 졸업과제에서 선정한 "URL 개수" 피처의 카운팅에서도 활용된다.

<품사처리>

품사 처리를 위해 Python의 자연어 처리 모듈인 NLTK(Natural Language Toolkit, 이하 NLTK)를 사용하여 토큰화와 품사 태깅을 진행하였다. 본 졸업과제에서 사용하는 품사 feature로는 명사, 대명사, 형용사, 동사, 부사 총 5가지의 품사만을 사용하므로 이를 위해 기존의 NLTK pos tag List에서 [그림2]와 같이 유사한 유형의 품사 리스트들을 Noun, Pronoun, Verb, Adjective, Adverb의 5가지 유형으로 통합시켰다.

```
if pos_tag in ['NN', 'NNS', 'NNP', 'NNPS', 'VBG']:
    new_tags.append((word, 'Noun'))
elif pos_tag in ['PRP', 'PRP$', 'WP', 'WP$']:
    new_tags.append((word, 'Pronoun'))
elif pos_tag in ['MD', 'VB', 'VBD', 'VBN', 'VBP', 'VBZ']:
    new_tags.append((word, 'Verb'))
elif pos_tag in ['JJ', 'JJR', 'JJS']:
    new_tags.append((word, 'Adjective'))
elif pos_tag in ['RB', 'RBR', 'RBS', 'WRB']:
    new_tags.append((word, 'Adverb'))
else:
    new_tags.append((word, pos_tag))
```

[그림 2] 품사 태그 리스트 처리 방법

<정규화>

각각의 데이터마다 그 길이가 다르고 피처의 분포가 다르기 때문에, 정확한 분석을 위해 min-max 정규화 함수를 정의([그림 3])하여 각 feature의 비율 값을 정규화하였다.

```
# Min-Max 정규화 함수 정의
def minmax_normalize(column):
    min_val = column.min()
    max_val = column.max()
    normalized_column = (column - min_val) / (max_val - min_val)
    return normalized_column
```

[그림 3] Min-Max 정규화 함수

2.3. 피처 분석

본 졸업과제의 목표인 새로운 유형의 스팸메일에 대한 대응과 다양한 유형(여러 개의 모달리티)으로 구성된 스팸메일의 분류를 위해 다중 입력 모델에 사용할 피처를 선정해야 한다. 이미지 스팸 데이터 유형에 포함된 텍스트를 추출하여 텍스트를 기반으로 각 모달리티별 차별점이 드러날 것이라 예상한 피처 15개([표 1])를 후보로 선정하였다.

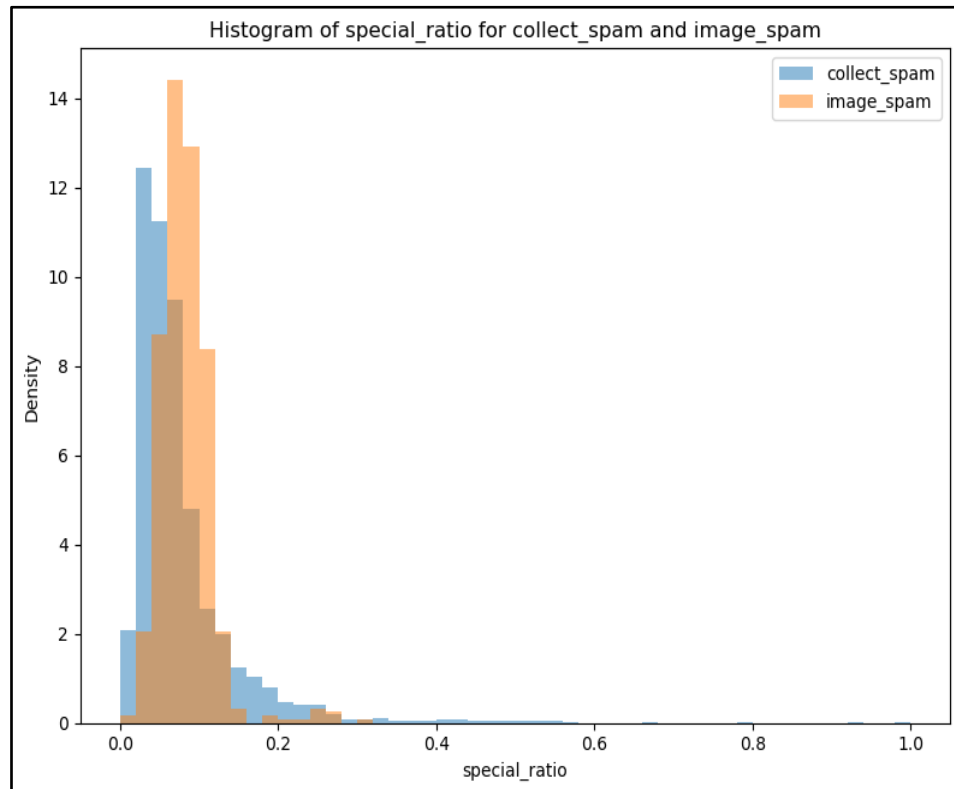
| | | | |
|---------------|---------------|---------------|---------------|
| 특수 문자 비율 | 숫자 비율 | URL 개수 | 대문자 비율 |
| 공백 비율 | 개행 문자 비율 | 명사 평균 비율 | 대명사 평균 비율 |
| 형용사 평균 비율 | 동사 평균 비율 | 부사 평균 비율 | 문장 내 단어 개수 평균 |
| 문장 내 문자 개수 평균 | 문단 내 단어 개수 평균 | 문단 내 문자 개수 평균 | 텍스트(TF-IDF) |

[표 1] 피처 후보

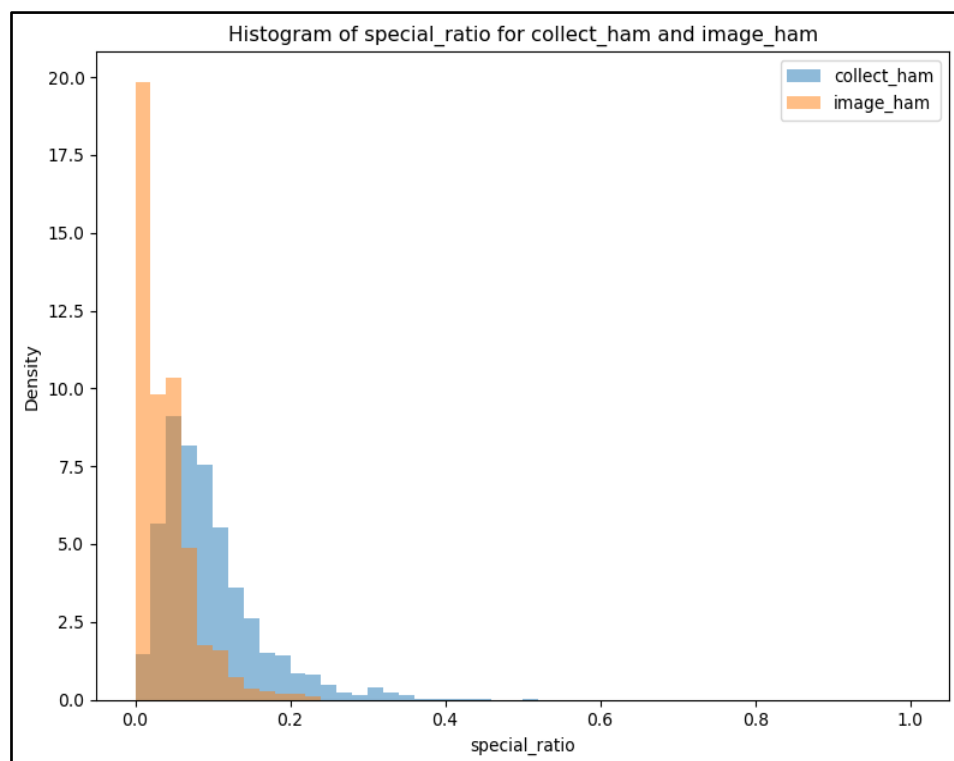
<피처 분석 방법>

피처를 분석하기 위해 각 데이터 셋에서의 피처들의 분포를 히스토그램으로 그려 차별점 여부를 확인하였다. 각 피처에 대해 모달리티간의 차별점이 선명하다면 해당 피처에 가중을 두고 모델 학습을 진행해야 다양한 유형의 새로운 패턴의 스팸메일에 잘 대응할 수 있기 때문에 차별점 여부를 기준으로 피처를 선정하였다. 아래는 선정된 피처 중 하나인 특수문자 비율을 수집 텍스트 데이터와 수집 이미지 데이터의 spam label에서의 분

포와 ham label에서의 분포를 히스토그램으로 그린 것이다.



[그림 4] 수집 텍스트와 수집 이미지 데이터의 spam label에서의 특수문자 비율 분포



[그림 5] 수집 텍스트와 수집 이미지 데이터의 ham label에서의 특수문자 비율 분포

<수집 + 이미지 멀티모달 피쳐 분석 및 선정>

수집 텍스트 데이터와 수집 이미지 데이터를 합친 후, 모든 피쳐마다 각 label(spam, ham)에서의 분포를 히스토그램을 통하여 확인하고 히스토그램상에서 overlapping 되는 면적(이 면적이 작을수록 특징의 차이가 극명하다는 것을 의미)을 계산하여 spam과 ham label에서 동시에 overlapping 면적이 0.5 이하로 나타나는 피쳐를 최종적으로 선정하였다. 다음은 수집 텍스트와 수집 이미지의 각 label에서의 overlapping 면적이다.

| | 수집+이미지(spam) | 수집+이미지(ham) |
|---------------------|--------------------|--------------------|
| special_ratio | 0.619257408 | 0.50994152 |
| number_ratio | 0.582159772 | 0.557551653 |
| url_count | 0.510525287 | 0.4040296 |
| upper_ratio | 0.361098234 | 0.153749418 |
| blank_ratio | 0.456401629 | 0.368186952 |
| crlf_ratio | 0.624672879 | 0.239462573 |
| Noun | 0.493328568 | 0.239271396 |
| Pronoun | 0.499256129 | 0.289794545 |
| Verb | 0.68065673 | 0.339623612 |
| Adjective | 0.524151542 | 0.389076109 |
| Adverb | 0.757173245 | 0.330230965 |
| avg_word_sentences | 0.642346522 | 0.553189585 |
| avg_char_sentences | 0.442354 | 0.509592957 |
| avg_word_paragraphs | 0.582450355 | 0.30979692 |
| avg_char_paragraphs | 0.634022932 | 0.299504705 |

[그림 6] 수집 텍스트와 수집 이미지 데이터의 overlapping 면적

<생성 + 이미지 멀티모달 피쳐 분석 및 선정>

위와 동일한 방법으로 생성 텍스트 데이터와 수집 이미지 데이터를 합친 뒤 히스토그램상에서 overlapping되는 면적을 계산하였다. 다음은 생성 텍스트와 수집 이미지의 각 label에서의 overlapping 면적이다.

| | 생성+이미지(spam) | 생성+이미지(ham) |
|---------------------|--------------------|--------------------|
| special_ratio | 0.527458788 | 0.493304871 |
| number_ratio | 0.320865888 | 0.355738076 |
| url_count | 0.447033034 | 0.518343747 |
| upper_ratio | 0.1480129 | 0.130628516 |
| blank_ratio | 0.772814351 | 0.306008086 |
| crlf_ratio | 0.400054735 | 0.284124516 |
| Noun | 0.579741564 | 0.179713874 |
| Pronoun | 0.316468297 | 0.172116825 |
| Verb | 0.706450695 | 0.335565043 |
| Adjective | 0.46197787 | 0.388980463 |
| Adverb | 0.644793187 | 0.305567022 |
| avg_word_sentences | 0.494451361 | 0.549831774 |
| avg_char_sentences | 0.471555175 | 0.523792021 |
| avg_word_paragraphs | 0.111520134 | 0.534824281 |
| avg_char_paragraphs | 0.106599339 | 0.556213068 |

[그림 7] 수집 텍스트와 수집 이미지 데이터의 overlapping 면적

2.4. 모델 구현 및 학습

2.4.1. 모델 선정

<MLP(Multi-Layer Perceptron)>

MLP는 가장 기본적인 인공신경망의 한 형태로 여러 개의 퍼셉트론 뉴런을 여러 층으로 쌓은 다층신경망 구조이며 입력층, 은닉층, 출력층으로 나누어져 있다. MLP는 인공신경망 종류 중 범주형 데이터 분류에 가장 적합하다고 알려져 있기 때문에 최종 선정하였다.

<LSTM(Long Short Term Memory)>

LSTM은 각 입력이 독립적인 MLP와 달리, 현재 입력에서 이전의 입력을 고려하는 방식이다. 또한 RNN에서 발전된 형태인 모델인 만큼 데이터의 길이가 길어질수록 초기 타임 스텝의 정보가 사라지는 단점을 보완하였다. 이러한 특징들 덕분에 LSTM은 주로 순차적인 데이터를 학습, 처리, 분류하는 데 주로 사용되고 감성 분석, 언어 모델링, 음성

인식, 비디오 분석 등에 주로 응용되고 있다. MLP와 LSTM중 어떤 모델을 사용할지 고려하는데 있어 둘의 성능을 직접 검증해 본 결과 별 차이가 없고, MLP가 이진 분류에 더욱 적합하기 때문에 LSTM은 고려 대상에서 제외하였다.

<SVM(Support Vector Machine)>

SVM은 데이터 분류에 전반적으로 많이 사용되는 모델이며 기존의 기계학습 기반 스팸 필터링에 많이 사용되는 그 성능이 검증된 알고리즘이다. 하지만, Spammer가 스팸 필터링을 우회하기 위하여 스팸메일에서 스팸 키워드의 철자를 의도적으로 잘못 표기하거나 다른 언어로 표기하는 등 다양한 방식의 우회기법에 대처하지 못한다는 단점이 있다. 본 졸업과제에서는 이러한 Concept drift 현상에 대처할 수 있는 Content based의 모델 구축을 목표로 하고 있기에 SVM 모델은 고려 대상에서 제외하였다.

2.4.2. MLP 모델 구현 및 학습

본 과제에서 분류기 모델은 총 5개로 분류된다. 이 모델들을 모두 합쳐 최종 모델을 구현하였다. 5개 분류기에서 사용되는 하이퍼 파라미터 튜닝과 관련한 내용은 2.4.3에 서술되어있다. 기본적인 단일 입력 MLP 모델의 구조는 [그림8]과 같다.

```
In [27]: def get_simple_model():
          model = Sequential()
          model.add(Dense(512, activation='relu', input_shape=(num_max,)))
          model.add(Dropout(0.5))
          model.add(Dense(256, activation='relu'))
          model.add(Dropout(0.5))
          model.add(Dense(1, activation='sigmoid'))
          model.summary()
          model.compile(loss='binary_crossentropy',
                        optimizer='adam',
                        metrics=['acc', keras.metrics.binary_accuracy])
          print('compile done')
          return model
```

[그림 8] MLP 단일 입력 모델 구조

다중 입력 MLP 모델의 구조는 [그림 9]와 같다.

```
In [20]: def get_simple_model():
# 은닉층과 출력층 설정
hidden_layer = Dense(64, activation='relu')(merged_input)
output_layer = Dense(1, activation='sigmoid', name='output')(hidden_layer)

# 모델 생성
model = Model(inputs=[text_input, special_char_input, number_count_input, url_count_input, upper_count_input, blank_count_input], outputs=output_layer)

# 모델 요약
model.summary()

# 모델 컴파일
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
print('compile done')
return model
```

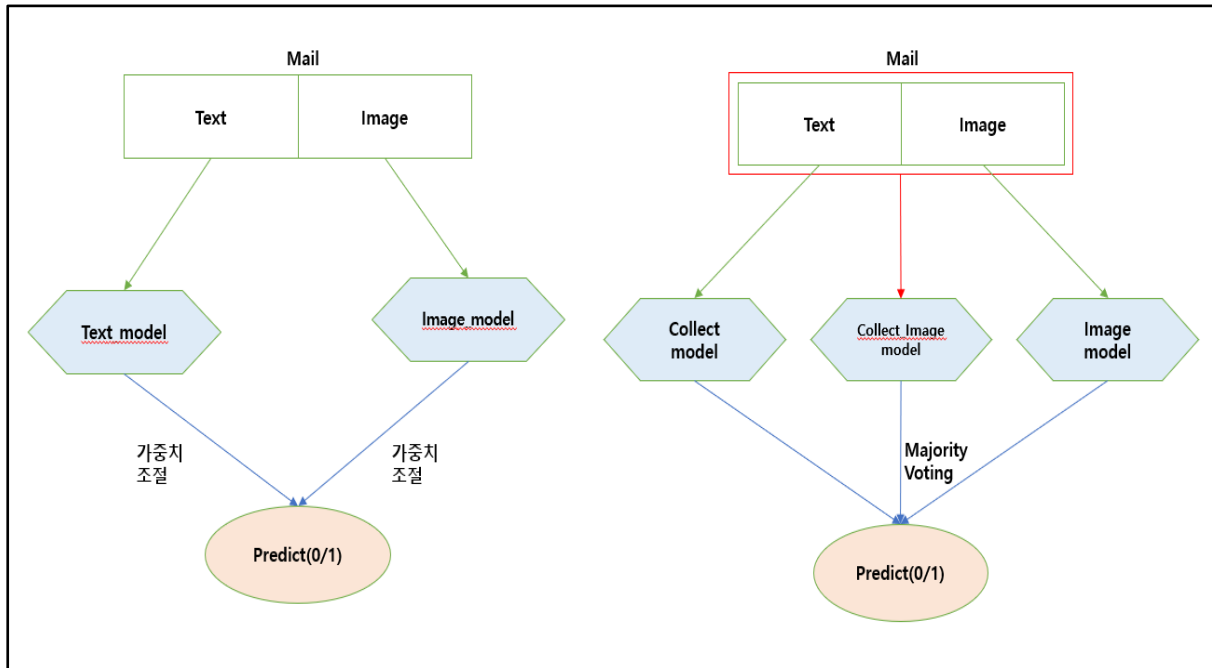
[그림 9] MLP 다중 입력 모델 구조

<최종 5 개 분류기 모델>

- 수집 데이터 TF-IDF 단일 피쳐 분류기(이하, collect model)
- 이미지 데이터 TF-IDF 단일 피쳐 분류기(이하, image model)
- 생성 데이터 TF-IDF 단일 피쳐 분류기(이하, generate model)
- 수집+이미지 데이터 TF-IDF 다중 피쳐 분류기(이하, collect_image model)
- 생성+이미지 데이터 TF-IDF 다중 피쳐 분류기(이하, generate_image model)

collect, image, generate model의 경우 TF-IDF 방식으로 tokenize된 가중치를 단일 피쳐로 사용한 모델이다. 즉, 각 모델은 각자가 학습한 데이터 셋에 특화된 모델이다. 그리고 collect_image model, generate_image model의 경우 앞서 피쳐 분석 과정을 통해 선정한 피쳐들을 사용하였다.

<최종 모델 시나리오>



[그림 10] 왼쪽(가중치를 조절하여 결과값 예측의 정확도를 높이는 경우), 오른쪽(과반수 투표)

[그림 10]에서 보면 원래는 이미지와 텍스트가 같이 동봉되어 메일이 전송된 경우 각각 이미지와 텍스트로 나누어 각 유형의 데이터로만 학습된 모델에 입력으로 넣고, 이후 나오는 예측 결과에 따라 가중치를 조절하여 정확도를 높이는 방향을 생각하였으나, 가중치를 조절하는 것보다 collect_image, create_image model을 따로 만들어 이 모델들이 가중치를 조절하는 역할을 하는 것(과반수 투표가 가중치 조절처럼 역할)이 더욱 성능향상에 도움이 된다고 예상했다. 즉, 다중 입력 모델을 다양한 유형의 데이터가 입력되었을 때 예측 결과를 결정하기 위한 최종 가중치처럼 사용한다. 최종 모델의 시나리오는 아래와 같다.

<최종 모델 시나리오 예제>

전제 조건 : 입력 시나리오는 수집, 생성, 이미지 단일로 들어오는 경우와 수집+이미지, 생성+이미지가 들어오는 경우다. 생성+수집이 들어오는 경우와 생성+수집+이미지가 들어오는 경우는 사실상 발신자 입장에서 거의 작성될 수 없는 상황이기에 제외했다.

입력 시나리오 1 : 단일 모달리티로 메일이 전송되는 경우에는 전송된 메일이 각 모달리티에 맞는 모델의 입력으로 들어가게 되고 최종 결과를 예측한다.

입력 시나리오 2 : 여러 개의 모달리티로 메일이 전송되는 경우(예를 들어, 이미지와 텍스트가 같이 동봉되어 전송되는 경우, [그림 6] 참고)에는 전송된 메일이 이미지와 텍스트로 나누어서 각각 단일모델의 입력으로 들어가고, 이미지와 텍스트가 합쳐진 입력이 통합 모델의 입력으로 들어가 3 개의 결과 값의 과반수 투표를 통해 최종 결과를 예측한다.

2.4.3. MLP 하이퍼 파라미터 튜닝

본 졸업과제에서 사용한 MLP의 하이퍼 파라미터 조합으로는 경우의 수가 너무 많아 성능 비교를 위한 전체 검증은 시간상 불가능하다고 판단하였다. 따라서, 각 하이퍼 파라미터의 특성에 근거하여 튜닝하였다.

본 졸업과제는 binary classification(이진 분류)를 기반으로 한다. 이진 분류의 경우 출력 노드가 1개이고, 이 node에서는 0~1 사이의 값을 가지면 마지막에 cast(0.5 이상이면 1, 미만이면 0)를 통해 1 또는 0의 값을 결과로 받을 수 있다. 따라서 활성화 함수로는 이진 분류에 적절한 함수인 시그모이드 함수를 선정하였다. 또한 손실 함수는 결과가 0 또는 1인 이진 분류 문제에서 사용되는 binary_crossentropy를 선정하였다. 마지막으로 최적화 함수로는 데이터를 학습할 때 학습 방향과 스텝 사이즈를 적절하게 결정해주는 Adam 방식을 선정하였다.

epoch의 경우 loss 값이 줄어들 때만 그 모델의 성능을 기록하고, 가장 loss 값이 작을 때에 epoch 값을 그래프로 그려 지정하였다. 이러한 방식으로 collect, image, generate, collect_image, generate_image model의 각 epoch는 순서대로 20, 30, 30, 30, 10이다.

2.5. 모델 성능 평가

성능 평가에 사용한 5개의 테스트 데이터 셋은 각 모델의 학습 전에 학습 데이터에서 5%씩 분할하여 별도로 저장한 데이터 셋으로 학습에는 사용되지 않은 데이터이며 각 모델을 5개의 테스트 데이터 셋으로 총 25번의 성능 평가를 진행하였다. 또한 각 모델의 학습은 Spam과 Ham의 비율을 1:1로 맞추어 진행하였기 때문에 정확도(Accuracy)를 기준으로 성능을 평가하였다.

다음은 TF-IDF 가중치를 이용해 Content-based 학습을 진행한 3가지 단일 입력 모델의 성능 평가 지표이다.

| | 수집 텍스트 | 생성 텍스트 | 수집 이미지 | 수집+이미지 | 생성+이미지 |
|---------------------------------|--------------|--------------|------------|--------|--------|
| Collect model (1feature) | 0.908 | 0.676 | 0.854 | 0.730 | 0.762 |
| Generate model (1feature) | 0.710 | 0.979 | 0.811 | 0.624 | 0.736 |
| Image model (1feature) | 0.620 | 0.743 | 1.0 | 0.739 | 0.734 |

[표 2] 단일 입력 모델의 정확도(Accuracy) 지표

수집 1feature 모델을 테스트 해 본 결과 수집 텍스트 테스트 셋의 정확도가 0.908로 가장 높은 것을 확인할 수 있다. 생성 1feature 모델의 테스트 결과도 생성 텍스트 테스트 셋의 정확도가 0.979로 가장 높았으며, 이미지 1feature 모델 역시 수집 이미지 테스트 셋의 정확도가 1.0으로 가장 높은 것을 확인하였다. 하지만 단일 유형의 데이터로 학습한 모델은 다른 유형의 데이터나 여러 유형의 데이터가 동시에 입력되는 경우에 정확도가 현저하게 떨어지는 것을 성능 평가를 통해 확인할 수 있다.

다음은 다양한 유형으로 구성된 메일의 분류를 위해 구축한 2가지 다중 입력 모델이다.

| | 수집 텍스트 | 생성 텍스트 | 수집 이미지 | 수집+이미지 | 생성+이미지 |
|-------------------------------------|--------------|--------------|--------------|--------------|--------------|
| Collect_image model (5features) | 0.889 | 0.842 | 0.914 | 0.915 | 0.884 |
| Generate_image model (6features) | 0.709 | 0.965 | 0.953 | 0.781 | 0.943 |

[표 3] 다중 입력 모델의 정확도(Accuracy) 지표

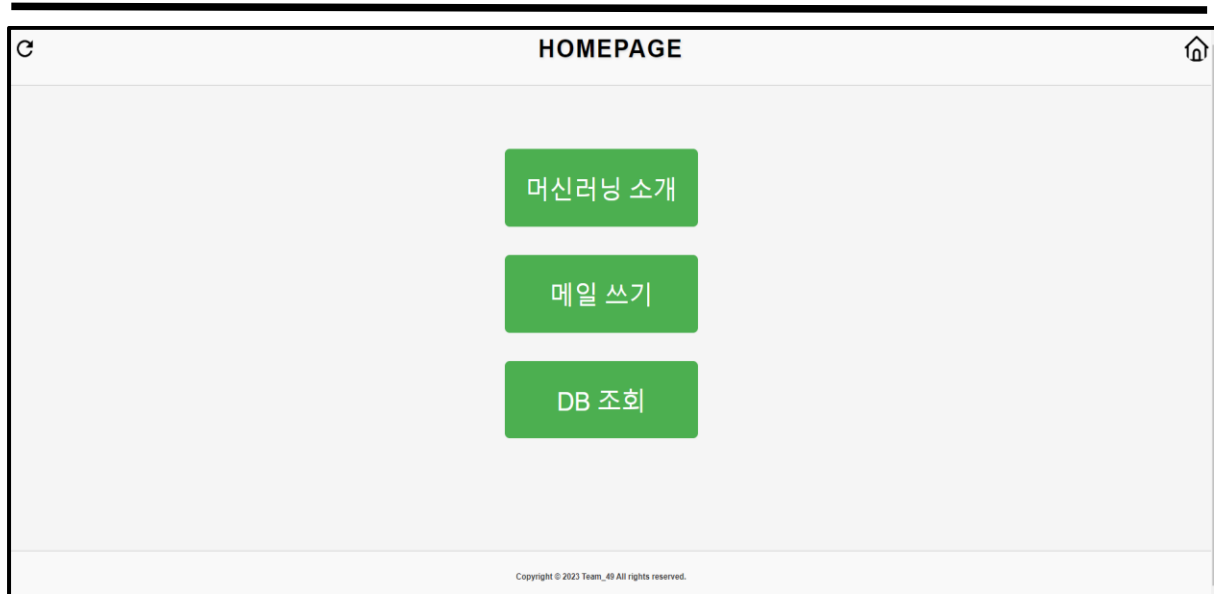
수집+이미지 5features 모델을 테스트 해 본 결과 수집+이미지 테스트 셋의 정확도가 0.915로 가장 높았으며, 수집 텍스트 테스트 셋과 수집 이미지 테스트 셋에서의 정확도는 각각 0.889, 0.914로 단일 입력 모델에 비해 유의미한 향상이 있었다.

생성+이미지 6features 모델의 테스트 결과 또한 생성+이미지 테스트 셋에서의 정확도가 0.965로 가장 높았으며, 마찬가지로 생성 텍스트 테스트 셋과 수집 이미지 테스트 셋에서의 정확도 역시 각각 0.965, 0.953으로 단일 입력 모델에 비해 좋아진 것을 확인하였다.

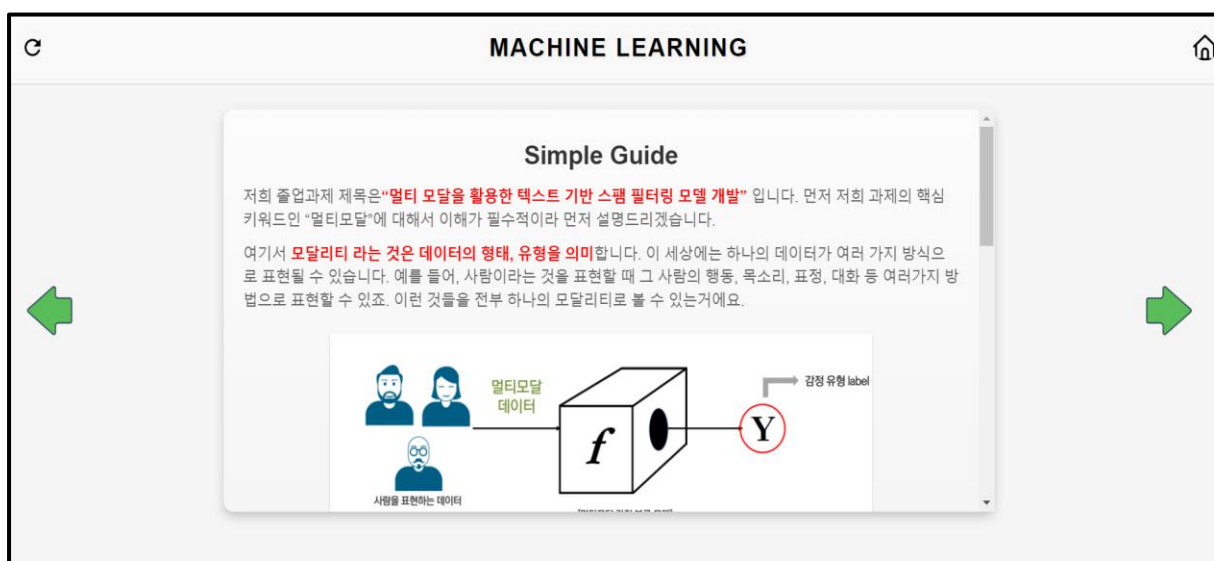
2.6. GUI 시각화 도구(WEB) 개발

2.6.1. Client-Side

클라이언트 사이드는 머신러닝 소개, 메일 쓰기, DB 조회 등 3가지의 선택지로 나뉜다. 간략한 머신러닝 소개를 구현, 화살표를 이용해 다음 페이지로 넘어갈 수 있으며, 더 이상 넘어갈 수 없을 시 alert를 띄운다.



[그림 11] 초기 화면



[그림 12] 머신러닝 소개

[생성 기반]

Chat-GPT로 생성한 스팸메일 200개와 햄 메일 200개를 선정해 클라이언트가 체험해 볼 수 있게 쿼리문을 이용해 DB에 집어넣는다. 그 결과 웹에서 GPT로 생성한 메일 기반의 생성 메일과 이미지 전송해 볼 수 있다.

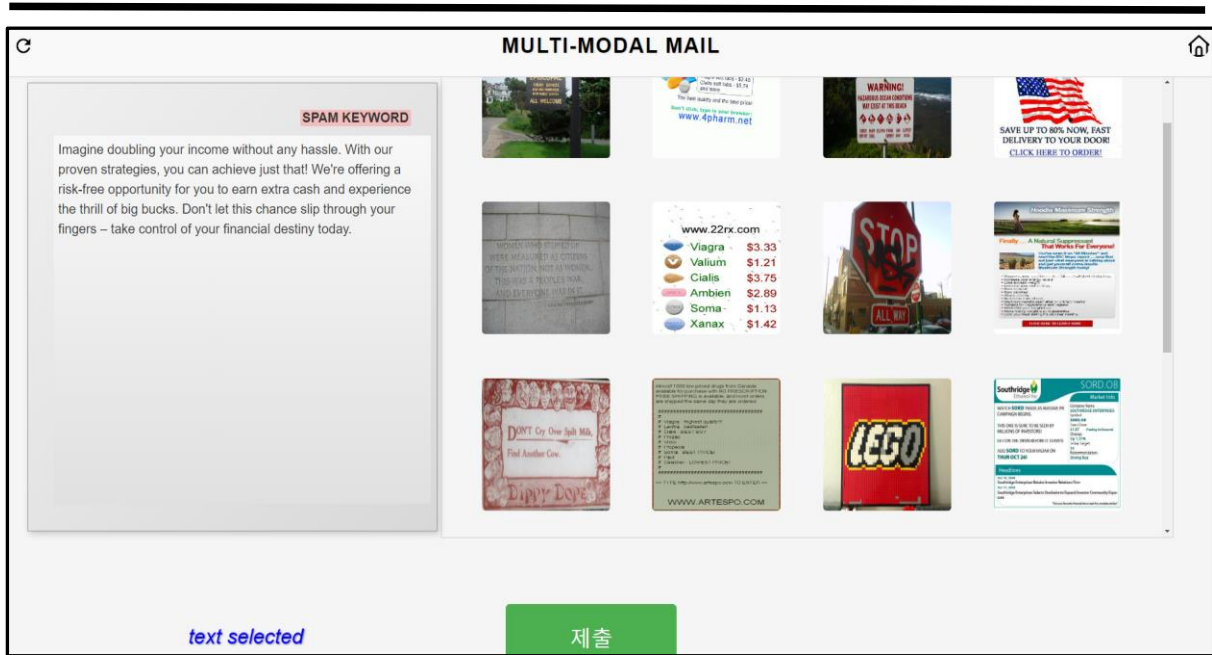
| idx | collect | generate | img | predict | feature | label |
|-----|---------|---|------|---------|---------|-------|
| 0 | NULL | Are you tired of living paycheck to paycheck? It's... | NULL | NULL | NULL | NULL |
| 1 | NULL | Imagine doubling your income without any hassle. W... | NULL | NULL | NULL | NULL |
| 2 | NULL | We're excited to announce that you're a winner! As... | NULL | NULL | NULL | NULL |
| 3 | NULL | Are you tired of living paycheck to paycheck? Our ... | NULL | NULL | NULL | NULL |
| 4 | NULL | We're excited to offer you a free gift that will t... | NULL | NULL | NULL | NULL |
| 5 | NULL | Imagine earning additional income without any limi... | NULL | NULL | NULL | NULL |
| 6 | NULL | Are you ready to take charge of your financial fut... | NULL | NULL | NULL | NULL |
| 7 | NULL | Congratulations! You've won a special cash bonus j... | NULL | NULL | NULL | NULL |
| 8 | NULL | Tired of dealing with debts and high interest rate... | NULL | NULL | NULL | NULL |
| 9 | NULL | Claim your free gift today and unlock the potentia... | NULL | NULL | NULL | NULL |
| 10 | NULL | Get ready to win big bucks with our exciting conte... | NULL | NULL | NULL | NULL |
| 11 | NULL | You're personally invited to a free consultation t... | NULL | NULL | NULL | NULL |
| 12 | NULL | Imagine doubling your income effortlessly. With ou... | NULL | NULL | NULL | NULL |
| 13 | NULL | Don't miss out on this miracle opportunity to earn... | NULL | NULL | NULL | NULL |
| 14 | NULL | Your ticket to financial success awaits! Unlock th... | NULL | NULL | NULL | NULL |
| 15 | NULL | Ready to embark on your journey to financial prosp... | NULL | NULL | NULL | NULL |
| 16 | NULL | You're invited to an exclusive opportunity to unlo... | NULL | NULL | NULL | NULL |
| 17 | NULL | Imagine a future where your financial worries are ... | NULL | NULL | NULL | NULL |
| 18 | NULL | Get ready to experience a miracle of financial abu... | NULL | NULL | NULL | NULL |
| 19 | NULL | Are you ready to seize your golden opportunity? Ou... | NULL | NULL | NULL | NULL |
| 20 | NULL | Are you tired of the 9-to-5 grind? It's time to be... | NULL | NULL | NULL | NULL |
| 21 | NULL | Imagine holding a ticket to unlimited extra income... | NULL | NULL | NULL | NULL |
| 22 | NULL | Miracles do happen, and our program is proof! Expe... | NULL | NULL | NULL | NULL |
| 23 | NULL | Your efforts deserve the best price, and our progr... | NULL | NULL | NULL | NULL |
| 24 | NULL | Double your income, double your joy! Our program o... | NULL | NULL | NULL | NULL |
| 25 | NULL | Are you ready to unlock your full earning potentia... | NULL | NULL | NULL | NULL |
| 26 | NULL | You've been granted exclusive access to a life of ... | NULL | NULL | NULL | NULL |

[그림 13] GPT 생성 데이터

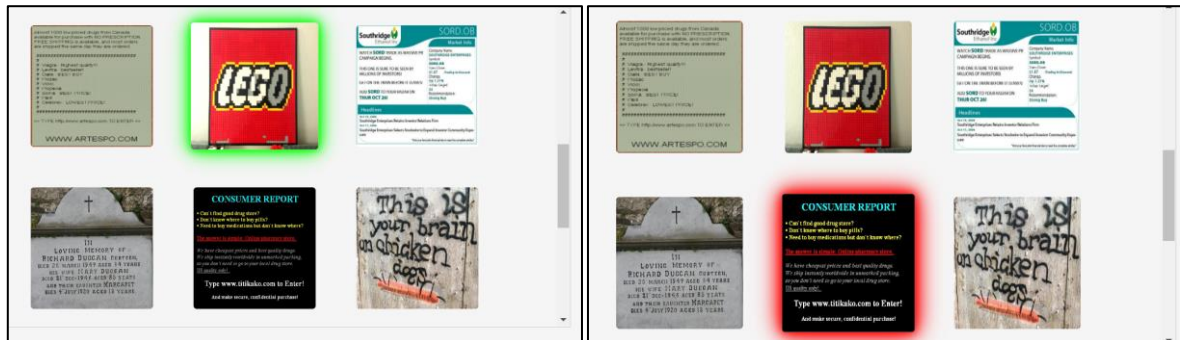
이미지들은 imageSrc 폴더 아래 번호순으로 저장되어 있으며 이것을 토대로 웹으로 flexbox로 불러온다.

왼쪽 생성 데이터 선택 시, 생성 메일은 메일 형식으로 출력되며 DB에 전송될 생성 데이터로 선택된다. 오른쪽 이미지 선택 시 JS Array에 저장된 이미지 내용이 선택된다. 왼쪽 생성 데이터 선택 시, 메일 형식으로 출력되고 오른쪽 이미지 선택 시 DB에 저장된 내용이 전송된다.

스팸이 아닌 메일은 마우스가 hover 시 네온 초록색으로, 스팸 메일은 네온 빨간색으로 표시된다. 다른 모달리티를 가진 메일을 각자 선택해 전송해 볼 수 있다.



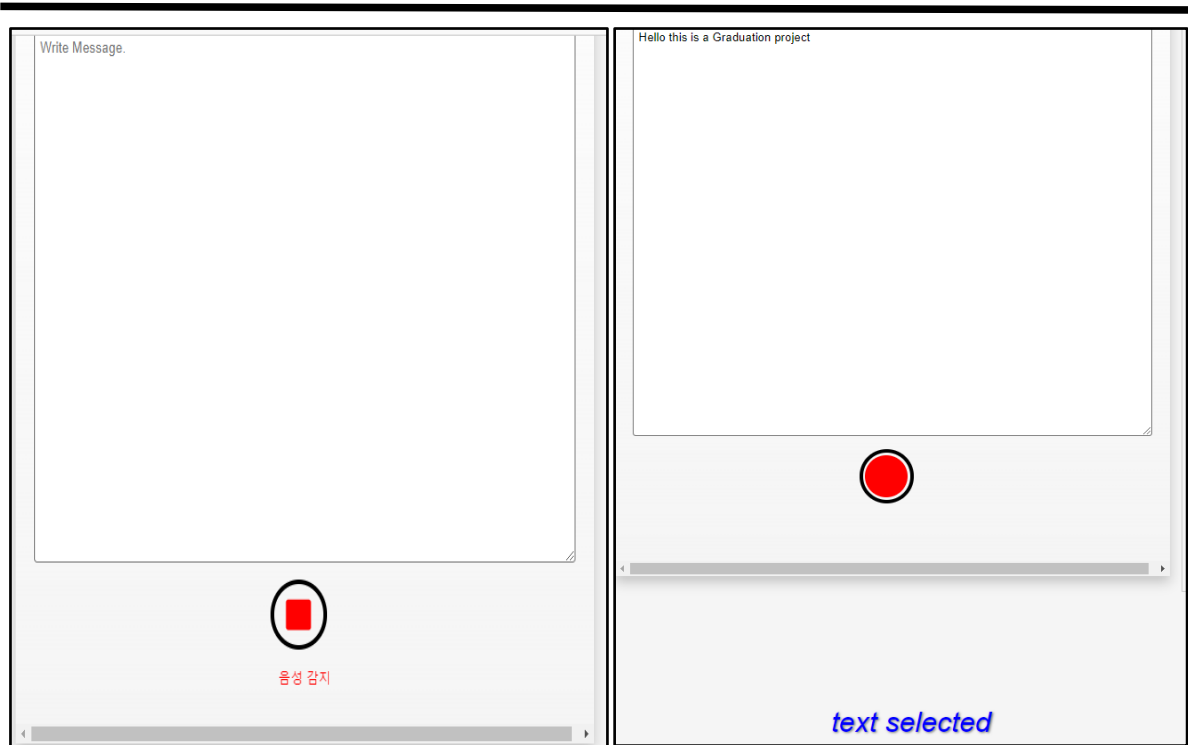
[그림 14] 생성 메일 선택



[그림 15] 이미지 선택 (좌 - Ham, 우 - Spam)

[텍스트 기반]

사용자의 입력을 받는 텍스트 기반으로도 전과 같이 입력할 수 있으며, 음성녹음 버튼을 통해 사람의 음성으로도 텍스트 인식이 가능하다.



[그림 16] 텍스트 선택 (좌 - 음성감지, 우 - 입력 후)

웹상에서 여러 개의 레이블로 나누어 DB에 집어넣으면 AI 모델에서 여러 가지 모달리티의 입력을 구별한다.

```
function getLabel(text,img,create){ //텍스트 0, 생성 1, 이미지 2, 텍스트 + 이미지 3, 생성 + 이미지 4
  var label = -1;
  if(text){ //텍스트가 존재하면
    if(img) label = 3;
    else label = 0;
  }
  else if(create){
    if(img) label = 4;
    else label = 1;
  }
  else if(img){
    label = 2;
  }
  return label;
}
```

[그림 17] 레이블 분류 (0 - 텍스트, 1 - 생성, 2 - 이미지, 3 - 텍스트+이미지, 4 - 생성 + 이미지)

데이터의 '를' '로' 바꿔주지 않으면 db에 전송되지 않으므로 텍스트 데이터를 전부 변

환해 주는 과정을 거쳐야 한다.

```
function convert(inputString) {  
    const resultString = inputString.replace(/'/g, "'");  
    return resultString;  
}
```

[그림 18] 텍스트 데이터 변환

2.6.2. Server-Side

<DataBase>

웹과의 연동을 위한 DB로 phpMyAdmin의 graduation_project database의 mail 테이블을 사용하였다.

[작성 메일 조회]

```
6 <?php  
7 $database = new mysqli("127.0.0.1", "root", "password", "graduation_project");  
8  
9 $data_arr = array();  
10 $result = mysqli_query($database, "SELECT * FROM mail");  
11  
12 while($row = mysqli_fetch_assoc($result)){  
13     $data = array("idx"=> $row['idx'], "collect"=> $row['collect'], "generate"=> $row['generate'], "img"=> $row['img'],  
14     "predict" => $row['predict'], "feature"=>$row['feature'], "label"=> $row['label']);  
15     $data_arr[] = $data;  
16 }  
17 mysqli_close($database);  
18  
19 ?>
```

[그림 19] mail 테이블의 모든 데이터를 \$data_arr 배열에 저장

php를 이용해 DB에 연결하고 select 쿼리문을 이용해 모든 table의 정보를 가져와 테이블 형식으로 웹에 출력하고, 각각의 row에 인덱스를 부여해, 클릭 시 row의 정보와 SPAM 키워드를 볼 수 있게 한다.

[생성 메일 조회]

| | | DATABASE | |
|----|---|--|---|
| 44 | Incredible deal It's effective Mail in order form Money back New domain extensions No cost Strong buy | | 0 |
| 45 | This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3 | ONE WAY STOP GPS DO NOT CITY OF CHICAGO 2005 ENTER ALL WAY | 1 |
| 46 | engine listings Search engines Sent in compliance Solution Special promotion Supplies are limited Talks about hidden charges Tells you it's an ad Terms and conditions The following form This isn't junk This isn't spam Trial We hate spam Web traffic Weekend getaway As seen on Bargain Be amazed | U.S. ★ DRUGS Licensed American Pharmacy NO PRESCRIPTION REQUIRED! SAVE UP TO 80% NOW, FAST DELIVERY TO YOUR DOOR! CLICK HERE TO ORDER! | 0 |
| 47 | This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3This is Test 1,2,3 | U.S. ★ DRUGS Licensed American Pharmacy NO PRESCRIPTION REQUIRED! SAVE UP TO 80% NOW, FAST DELIVERY TO YOUR DOOR! CLICK HERE TO ORDER! | 1 |
| 48 | Solution Special promotion Supplies are limited Talks about hidden charges Tells you it's an ad Terms and conditions The following form This isn't junk This isn't spam Trial We hate spam Web traffic Weekend getaway As seen on Bargain Be amazed Bonus Buy direct Buying judgments Call Call free Can't live without Cheap Claims not to be selling anything Clearance Click to remove Collect Compare rates Complete for your business Deal Don't hesitate! Explode your business F r e e Fantastic deal For Only Free DVD Free hosting Get it now Guarantee Guaranteed Have you been turned down? Increase sales Increase your sales Incredible deal It's effective Mail in order form Money back New domain extensions No cost Strong buy | U.S. ★ DRUGS Licensed American Pharmacy NO PRESCRIPTION REQUIRED! SAVE UP TO 80% NOW, FAST DELIVERY TO YOUR DOOR! CLICK HERE TO ORDER! | 1 |

[그림 20] mail 테이블 시각화

<SPAM 키워드 조회>

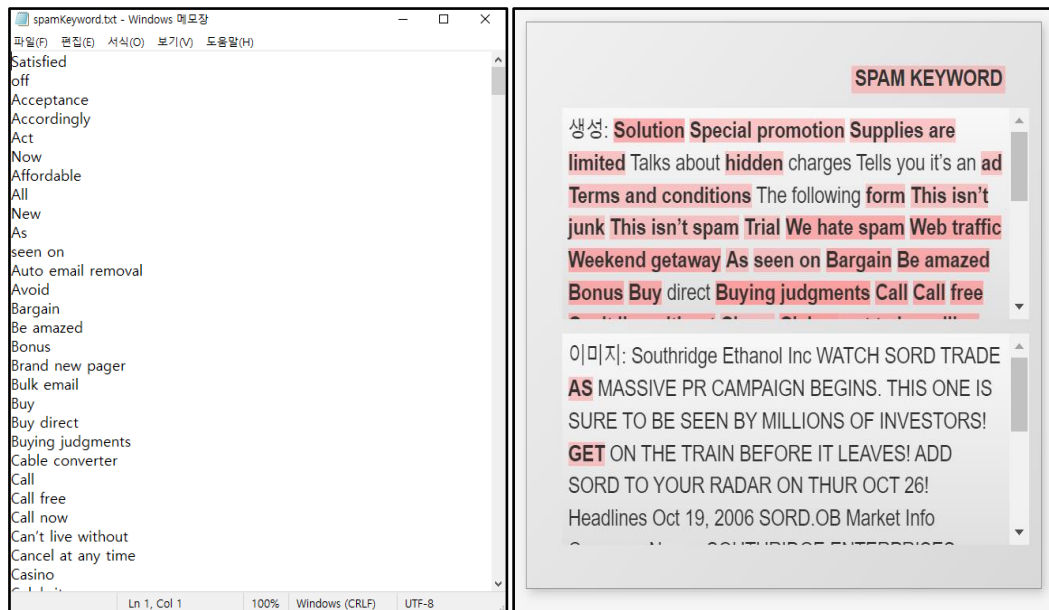
출력된 행의 메시지 내용을 클릭하면 이벤트가 트리거 되며, 스팸 키워드를 각자의 모달리티에 맞게 하이라이팅하여 조회할 수 있다.

| SPAM KEYWORD | SPAM KEYWORD |
|--|----------------------|
| <p>생성: engine listings Search engines Sent in compliance Solution Special promotion Supplies are limited Talks about hidden charges Tells you it's an ad Terms and conditions The following form This isn't junk This isn't spam Trial We hate spam Web traffic Weekend getaway As seen on Bargain Be amazed</p> | <p>유저: hey hello</p> |

[그림 21] 메시지 이벤트 트리거

spamKeyword.txt 에서 '₩n'를 기준으로 모든 문자를 잡아 spam 배열에 push 하여 스

팸 키워드를 전부 JS에 적을 필요 없이 구현하였다.



[그림 22] 스팸 키워드 탐지

2.7. 스팸 필터링 모델과 웹 연동

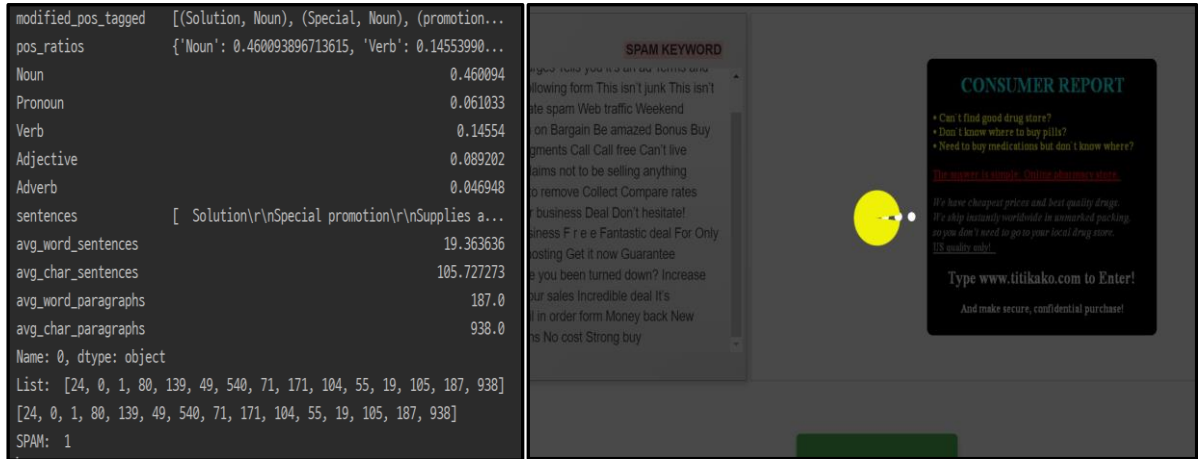
2.7.1. Polling

웹에서 메일 데이터가 입력되면 Python에서 2초 주기로 감지한다. 새로운 데이터를 찾았다면 update 쿼리문을 이용해 메일을 분석해 spam여부와 feature들을 추출해 JSON 형식으로 변환하여 추가한다.

```
833 last_id = max(row['idx'] for row in new_rows) # last_id 최신으로 업데이트
834 lst_json = json.dumps(lst_feature) # 리스트를 JSON 형식으로 변환
835 print(lst_feature)
836 print("SPAM: ", result)
837 query = f"UPDATE mail SET predict = %s, feature = %s WHERE idx = %s"
838 data = (result, lst_json, last_id)
839 cursor.execute(query, data)
840
841
842 time.sleep(2) # 2초 간격으로 Polling
843 connection.commit() # 실시간 db 연동 (중요!) 빠지 말 것
```

[그림 23] Polling방식으로 DB의 변화 감지

웹상에서 전송되는 데이터 [그림 24]를 보면 팩맨 애니메이션과 함께 다른 모달리티의 두 데이터가 같이 전송되는 것을 볼 수 있다. 스팸 필터링 모델이 웹에서의 입력을 감지하고, 분석한 데이터를 DB의 칼럼에 업데이트한다. [그림 25]는 스팸 피쳐들과 스팸 여부가 row에 입력된 결과이다. 그 후에, 웹에서 spam 메시지 여부와 JSON 형식으로 변환된 메일의 스팸 피쳐들을 웹으로 가져온다.



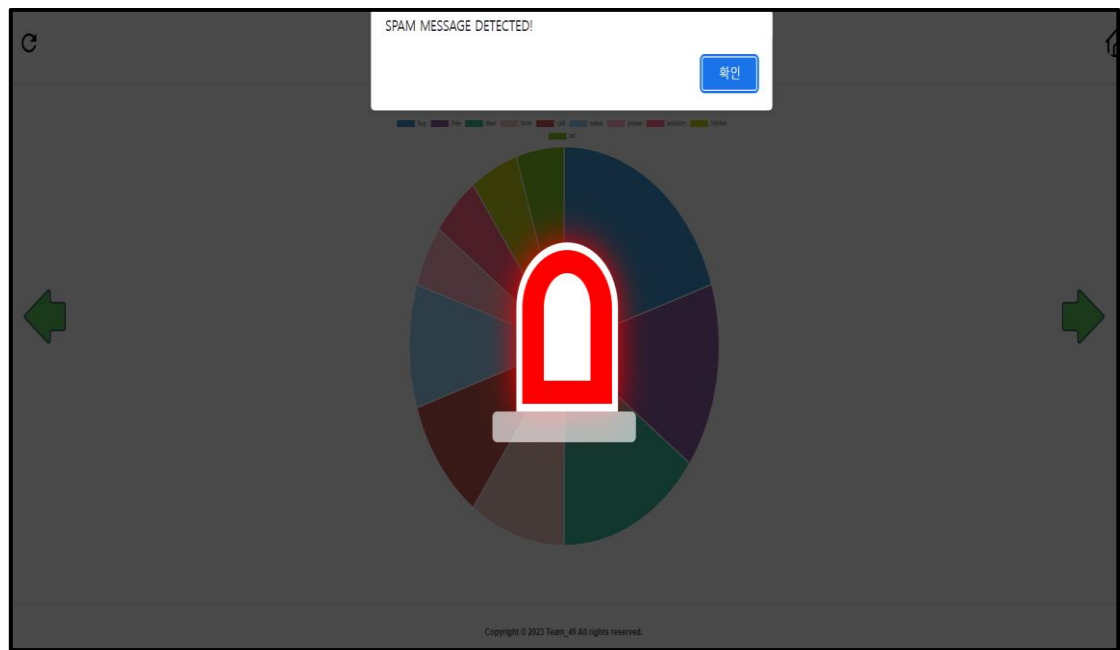
[그림 24] 웹에서 보낸 데이터를 AI모델에서 분석 후 업데이트

| | | | |
|------------------------|---------------------------------------|---------------------------------------|---|
| Solution | CONSUMER REPORT • Cant find good drug | [24, 0, 1, 80, 139, 49, 540, 71, 171, | 4 |
| Special promotion | store? Dont ... | 104, 55, 19,... | |
| Supplies are limite... | | | |

[그림 25] 업데이트 된 테이블 ROW

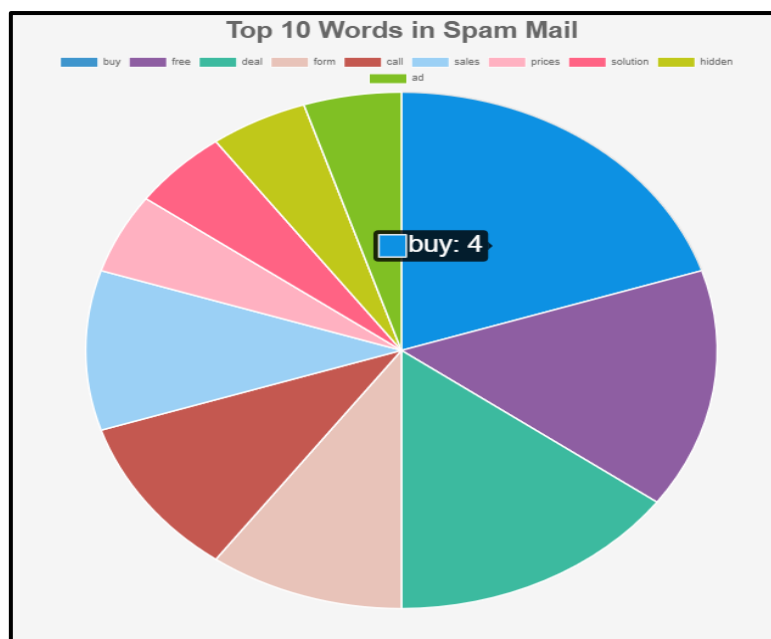
2.7.2. Chart.js를 이용한 SPAM 판독과 시각화

스팸 필터링 모델이 메일의 사용자의 입력을 스팸 메시지로 판단하였을 경우, 3초간 사이렌 애니메이션을 화면에 출력한다.



[그림 26] 스팸 메일로 감지된 경우 사이렌 애니메이션

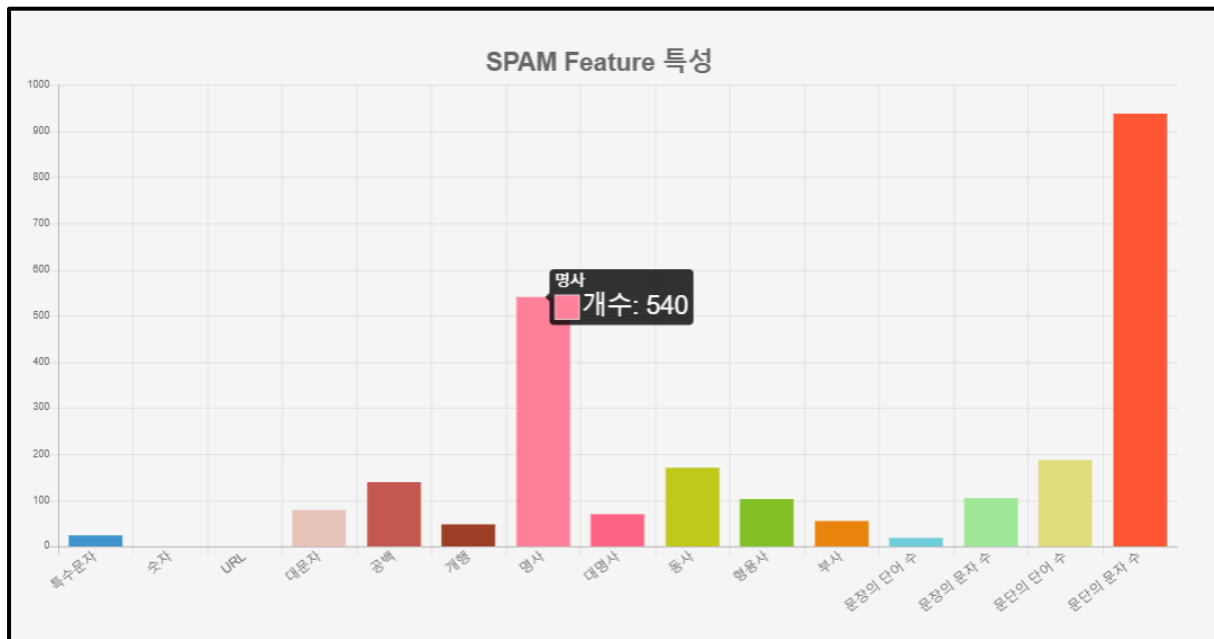
로컬 스토리지로 보낸 생성 메일, 이미지, 작성 메일들을 뽑아 전부 이어 붙인 뒤, 단어만을 추출해 spamKeyword 배열 안에 있는 요소와 같은 것이 있다면 associative 배열 추가 후 하나씩 카운팅한 후에, 내림차순 정렬하여 상위 10개를 그래프에 출력하였다.



[그림 27] 최대 10개 키워드의 Pie 차트

Chart.js의 그래프들은 화살표로 이동이 가능해 Pie-chart와 Bar-chart를 사용자의 기호에 따라 조회할 수 있다.

스팸 필터링 모델에서 읽어온 JSON 리스트를 Parsing해 JavaScript Array로 변환 후, Chart.js의 bar-chart의 데이터로 사용한다.



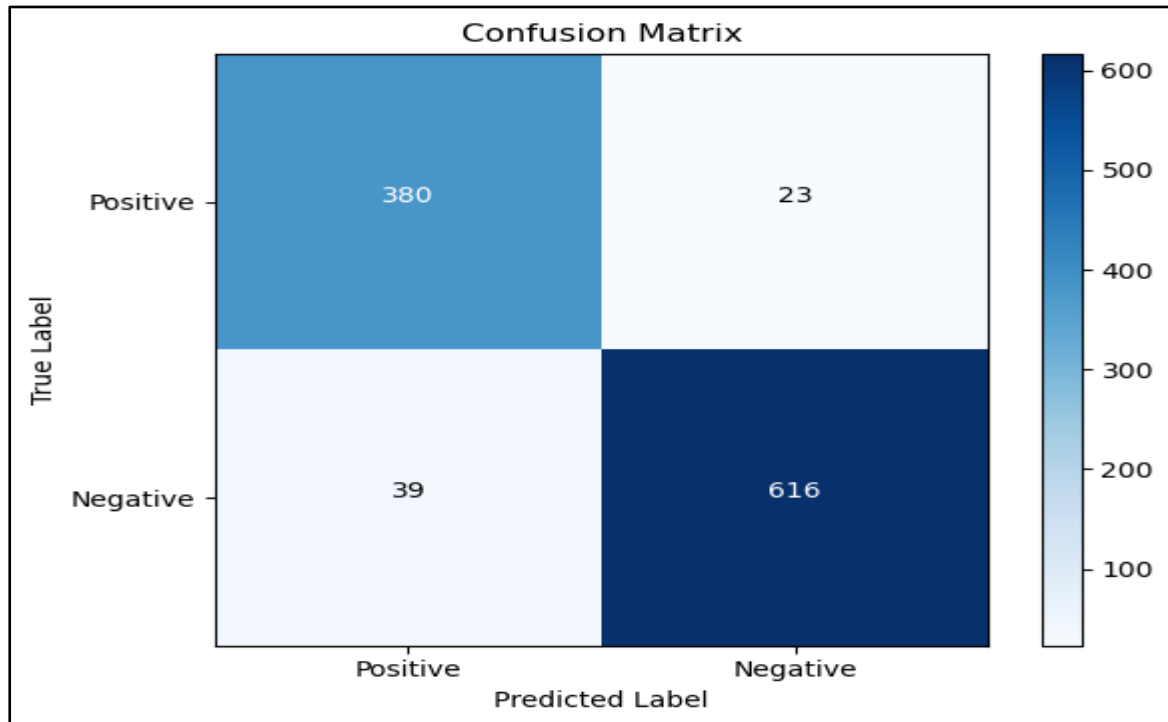
[그림 28] 피쳐 분석 결과 바그래프

<동작 영상 링크>

https://drive.google.com/file/d/1JhfkuglrYs5WLyjM6XIDDEk3_KbBv84/view?usp=drive_link

3. 연구결과 분석 및 평가

다음은 최종 구현한 멀티모달 스팸 필터링 모델의 테스트 결과로 생성한 Confusion Matrix이다.



[그림 29] 최종 멀티모달 스팸 필터링 모델의 Confusion Matrix

<최종 멀티모달 스팸 필터링 모델의 성능 지표>

| | |
|-------------------|--------|
| Accuracy (정확도) | 0.9414 |
| Recall (재현율) | 0.9069 |
| Precision (정밀도) | 0.9429 |
| Specificity (특이도) | 0.9640 |
| F1 Score | 0.9246 |

해당 결과를 바탕으로 실제 값이 Spam인 데이터를 Spam이라고 바르게 예측한 정도인 Recall을 계산해 본 결과 0.9069로 다른 요소에 비해 낮은 값을 보였다. 실제로 Spam이 아닌 메일을 Spam이라고 분류하게 되면 사용자의 불편함을 유발할 수 있기 때문에 개선이 필요해 보인다.

결과적으로 본 졸업과제의 최종 모델이 다양한 유형이 합쳐진 메일의 스팸 필터링에서 개선된 성능을 보이는 것을 확인하였다. 하지만, Recall(재현율)은 간신히 90%에 머무는 것으로 보아 더욱 다양한 데이터셋을 분석하여 피처의 다양성을 확보한다면 더욱 성능을 개선할 수 있을 것으로 보인다. 데이터 셋의 다양성을 높일 수 있는 현실적인 방법을 찾고, 더 많은 데이터를 수집하면 최종 성능은 더 높아질 것으로 예측한다.

4. 결론 및 향후 연구 방향

본 졸업과제에서는 멀티모달을 활용한 텍스트 기반 스팸 필터링 모델을 개발하였다. 이 과정에서 각 모달리티에 대한 피처 분석을 진행하였고 본 과제에서 분석한 피처가 다양한 유형의 메일에 대한 분류 정확도를 높이는 것을 확인했다. 기존에 잘 알려진 스팸 메일의 특징인 URL, 특정 키워드, 공백 수, 개행 문자 수 등의 feature들 외에도 직접 선정한 품사나, 문단 내 단어 수 비율 등과 같은 feature들을 분석하여 적용해 다양한 유형의 메일에 대처가능한 멀티모달에 대한 정확도를 높이하고자 하였다.

더욱 많은 피처를 분석하고 선정하고 싶었지만, 텍스트 기반이다 보니 각 모달리티에 대한 극명한 차이를 나타내는 피처가 많이 없던 점, 분석할 데이터 셋의 충분하지 못한 양 등의 이유로 본 졸업과제에서 선정한 피처 후보 15개만 분석해 볼 수 있었다. 음성 데이터에 대한 모달리티 처리를 통해 다양성 대처에 대한 활용도를 높이하고자 했으나, 영문 음성 데이터 셋 자료 수집에 어려움이 있어 이는 모달리티로 추가하지 못하였다.

향후에는 더욱 시간을 들여 다양한 데이터 셋을 수집하고, 메일함에서 일정 기간동안 직접 모은 다양한 형태의 메일들을 활용하여 더욱 다양한 모달리티를 추가하여 피처를 분석해 학습시킨다면 더 많은 유형의 데이터에 대한 대처가 가능한 모델을 개발할 수 있을 것으로 예상된다. 또한, 텍스트 기반과 더불어 음성과 이미지 데이터 셋을 텍스트로 변환하기전의 feature들을 이용한다면 더욱더 높은 성능을 가질 수 있을 것으로 예상된다.

5. 구성원별 역할 및 개발 일정

| 이름 | 역할 |
|-----|---|
| 윤상호 | 데이터 수집 및 정리, 데이터 생성, 전처리(중복제거, 토큰화, 품사처리), 분석용 feature 15개 생성, feature 분석, 단일 입력 스팸 필터링 모델 개발, 멀티모달 스팸 필터링 모델 개발 |
| 조재홍 | 데이터 수집 및 정리, 데이터 생성, 전처리(중복제거, 토큰화, 정규화), 분석용 feature 15개 생성, feature 분석, 단일 입력 스팸 필터링 모델 개발, 멀티모달 스팸 필터링 모델 개발 |
| 이강우 | 데이터 수집 및 정리, 데이터 생성, 클라이언트 사이드(HTML, JavaScript, CSS)로 클라이언트가 이용할 수 있는 웹페이지 구현, 서버 사이드(XAMPP, PHP)를 이용해 DB를 구축해 서버와 웹, AI 모델과의 데이터 연동 |

| 6월 | | | | 7월 | | | | 8월 | | | | 9월 | | | |
|--|----|--------------|----|---------------------|----|----------|----------------|-----------------|----|-------------------|----|----|----|----|----|
| 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 |
| 스터디(멀티 모달 AI, 스팸 필터링 기법, MLP, LSTM, SVM) | | | | | | | | | | | | | | | |
| | | 텍스트 수집 및 전처리 | | | | | | | | | | | | | |
| | | | | 모델 선정 및 단일 입력 모델 구축 | | | | | | | | | | | |
| | | | | | | 중간보고서 작성 | | | | | | | | | |
| | | | | | | | 이미지 수집, 텍스트 생성 | | | | | | | | |
| | | | | | | | | feature 분석 및 선정 | | | | | | | |
| | | | | | | | | | | 멀티모달 스팸 필터링 모델 개발 | | | | | |

| | | | | | | | | | | | | | | |
|--|--|--|--|--|-------------------|--|--|--|--|--|--|--------------------|---------------|--|
| | | | | | GUI 구축, 데이터베이스 구축 | | | | | | | | | |
| | | | | | | | | | | | | 웹, DB 개발, 모델 연동 | | |
| | | | | | | | | | | | | | 최종 보고 서 작성 | |

6. 참고 문헌

- [1] Karim, Asif, et al. "A comprehensive survey for intelligent spam email detection." *IEEE Access* 7 (2019): 168261-168295.
- [2] 김종욱 and 최미정. (2022). BiLSTM 기반의 악성 파일 자동 탐지 및 분류 방안 연구. *KNOM Review*, 25(1), 37-48.
- [3] A. S. Katasev, L. Y. Emaletdinova and D. V. Kataseva, "Neural Network Spam Filtering Technology," *2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, Moscow, Russia, 2018, pp. 1-5, doi: 10.1109/ICIEAM.2018.8728862.
- [4] Dada EG, Bassi JS, Chiroma H, Abdulhamid SM, Adetunmbi AO, Ajibuwa OE. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*. 2019 Jun 10;5(6):e01802. doi: 10.1016/j.heliyon.2019.e01802. PMID: 31211254; PMCID: PMC6562150.
- [5] 유경탁. "인공신경망을 이용한 스팸 메시지 탐지 기법." 국내석사학위논문 숭실대학교 소프트웨어특성화대학원, 2016. 서울
- [6] 김진우, 조혜인 and 이봉규. (2019). 인공신경망을 적용한 악성 댓글 분류 모델들의 성능 비교, *디지털콘텐츠학회논문지*, 20(7), 1429-1437.