YouTube Spam Comments Analysis & Deletion Recommendation Service

Team 1

19102076 Minsoo Kang
19102082 Jaeyoung Kim
19102084 Hyeok Kim

https://blog.naver.com/minimine_95

Table of Contents

- Overview
- Goal of this project
- Getting YouTube comment API
- Pipeline
- Processing / Applied model
- Storing the processed Data
- Web process & design

YouTube KR

Overview

Comments on videos in YouTube

The negative effects of many spam comments

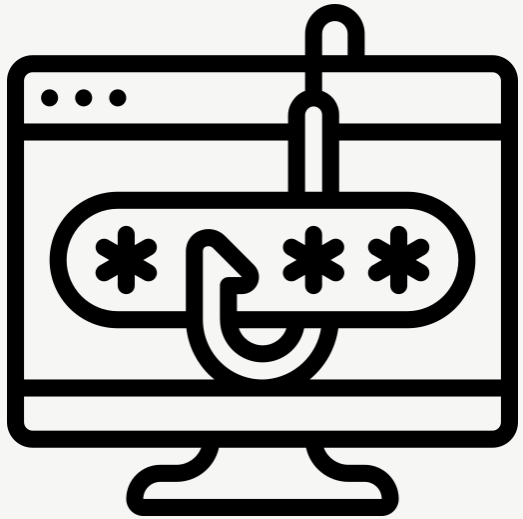
↓

Users & Securities

https://blog.naver.com/minimine_95

YouTube KR

Overview



https://blog.naver.com/minimine_95

YouTube KR

Goal of this project

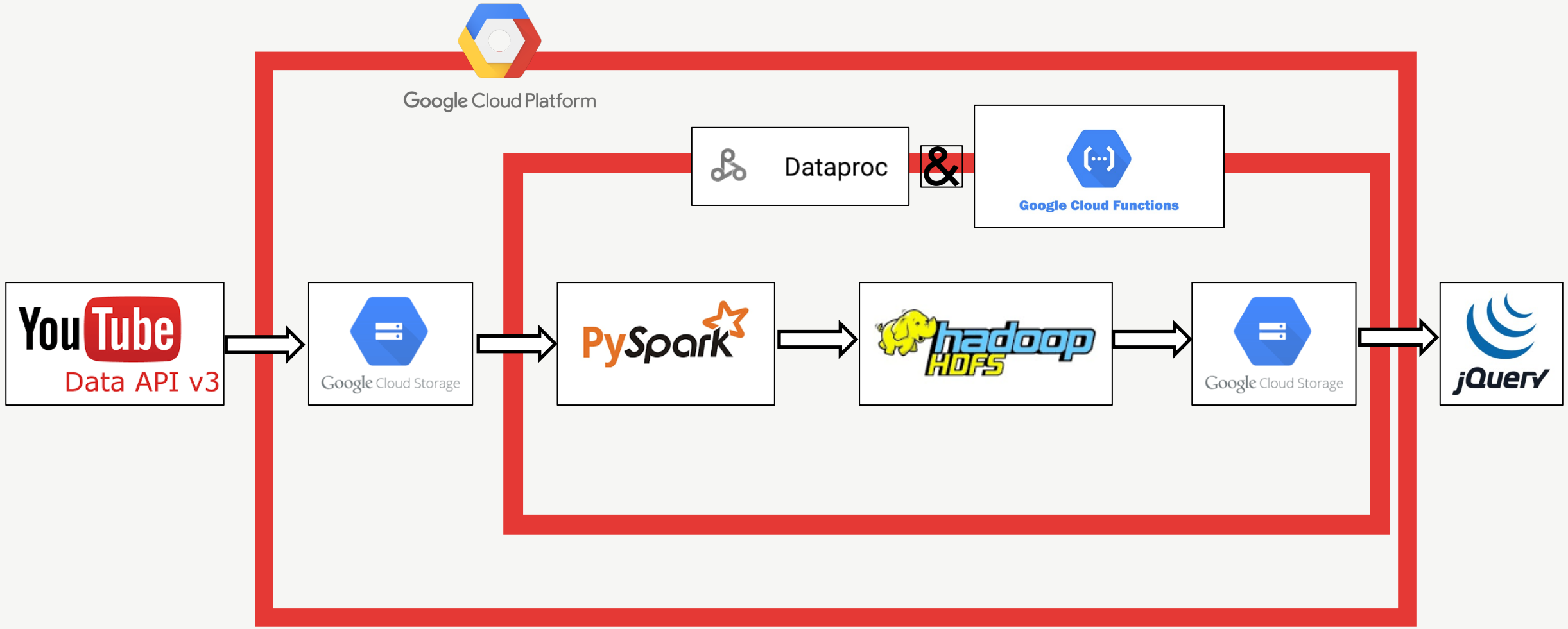
Detecting the spam comments in YouTube

Prevent the damages from frauds

https://blog.naver.com/minimine_95

YouTube KR

Pipeline



https://blog.naver.com/minimine_95

Getting YouTube v3 API

```
import pandas as pd
from googleapiclient.discovery import build
import time
import os
import datetime

# gcs
from google.cloud import storage

os.environ["GOOGLE_APPLICATION_CREDENTIALS"]="db-project-381514-6612cc127860.json"


video_id = 'Os_heh8vPfs'
comments = list()
api_obj = build('youtube', 'v3', developerKey='AIzaSyB0yxCBhji8ygtCWUDaxs7wV6hdVcy5gR0')


# GCS 버킷 설정
bucket_name = 'dbproject_comment'  # GCS 버킷 이름
blob_name = 'comment_data'  # GCS 버킷에 저장될 Blob 이름
```

Extract the required information and make DataFrame

```
while True:
    response = api_obj.commentThreads().list(part='snippet,replies', videoId=video_id, maxResults=100).execute()
    new_comments = []

    for item in response['items']:
        comment = item['snippet']['topLevelComment']['snippet']
        if [comment['textDisplay'], comment['authorDisplayName'], comment['publishedAt'], comment['likeCount']] not in comments:
            new_comments.append([comment['textDisplay'], comment['authorDisplayName'], comment['publishedAt'], comment['likeCount']])

        if item['snippet']['totalReplyCount'] > 0:
            for reply_item in item['replies']['comments']:
                reply = reply_item['snippet']
                if [reply['textDisplay'], reply['authorDisplayName'], reply['publishedAt'], reply['likeCount']] not in comments:
                    new_comments.append([reply['textDisplay'], reply['authorDisplayName'], reply['publishedAt'], reply['likeCount']])

    if new_comments:
        df = pd.DataFrame(new_comments, columns=['댓글','작성자','작성시간','좋아요수'])
        print(df)
        comments += new_comments
```

Send DataFrame to google cloud storage based on current time

```
# GCS 클라이언트 생성
client = storage.Client()

# 존재하는 버킷의 이름
bucket_name = 'dbproject_comment'

# 폴더 경로와 현재 시간을 포함한 파일 이름 생성
current_time = datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
folder_name = 'comment_output'
blob_name = f"{folder_name}/{current_time}.csv"

# 데이터프레임을 GCS 버킷에 업로드
csv_string = df.to_csv(index=False)

# Blob 객체 생성
blob = client.bucket(bucket_name).blob(blob_name)

blob.upload_from_string(csv_string.encode(), content_type='text/csv')

print(f'Dataframe uploaded to {bucket_name}/{blob_name}.')
```

https://blog.naver.com/minimine_95

Load the model to use and used the 'Roberta base' model using 'Bert'

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

tokenizer = AutoTokenizer.from_pretrained("mariagrandury/roberta-base-finetuned-sms-spam-detection")

model = AutoModelForSequenceClassification.from_pretrained("mariagrandury/roberta-base-finetuned-sms-spam-detection")
```

"Ok lar... Joking wif u oni... "	0 (ham)
"Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's "	1 (spam)
"U dun say so early hor... U c already then say... "	0 (ham)
"Nah I don't think he goes to usf, he lives around here though "	0 (ham)
"FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, £1.50 to rcv "	1 (spam)
"Even my brother is not like to speak with me. They treat me like aids patent. "	0 (ham)
"As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends…	0 (ham)
"WINNER!! As a valued network customer you have been selected to receivea £900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only…	1 (spam)

https://blog.naver.com/minimine_95

Eliminating missing values and non-terminal terms

```
#결측치제거
df = df.dropna()
# "댓글" 컬럼에서 이모티콘과 특수문자 제거
df = df.withColumn("댓글", regexp_replace(col("댓글"), "[^\uAC00-\uD7A3xfe0-9a-zA-Z\\s]", "").cast(StringType()))
```

By specifying udf as a function, it allows for distributed processing

```
#udf 선언
def predict_spam_udf(text):
    # 전처리
    inputs = tokenizer(text, return_tensors="pt", padding=True, truncation=True)
    # 예측
    outputs = model(**inputs)
    # 결과 반환
    return int(torch.argmax(outputs.logits)), float(torch.softmax(outputs.logits,dim=1)[0][1])

# 구조체 선언
predict_spam = udf(predict_spam_udf, StructType([
    StructField("prediction", IntegerType()),
    StructField("probability", FloatType())
]))
```

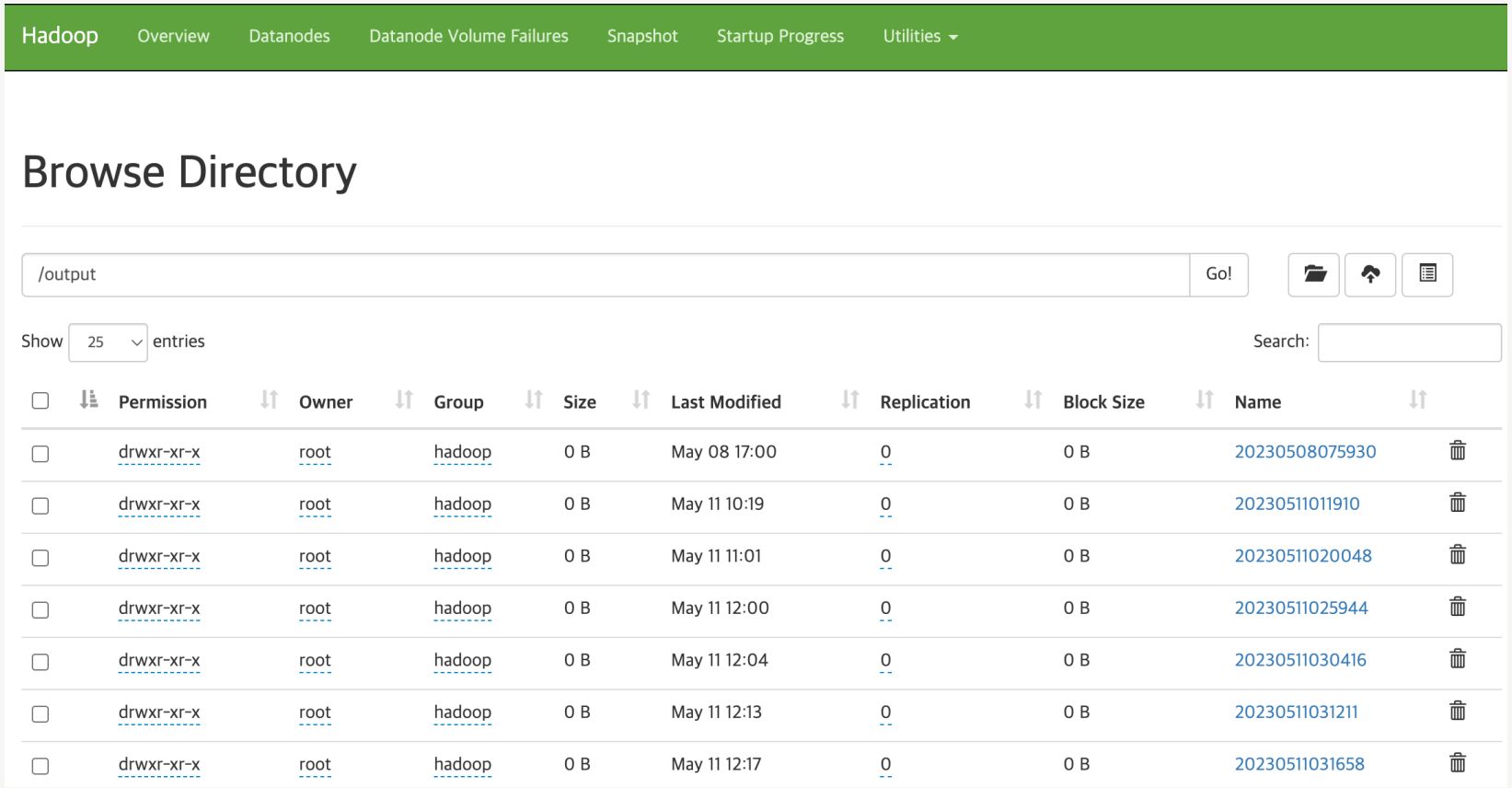
https://blog.naver.com/minimine_95

Data to which pre-processing and models are applied are stored in Hadoop and GCS

```
#  파일을 저장하려는 경로 지정
output_path_spam = f"gs://dbproject_comment/spam_output/{file_name}"


# 데이터프레임을 CSV 파일로 저장
spam_df.write.option("header", "true") \
        .option("delimiter", ",") \
        .option("quote", "\"") \
        .option("escape", "\"") \
        .csv(output_path_spam)

# 스팸 데이터를 csv로 past_data에 저장

#  파일을 저장하려는 경로 지정
output_path_process = f"gs://dbproject_comment/processing_output/{file_name}"


df.write.option('header', 'true') \
        .option('delimiter', ',') \
        .option('quote', '"') \
        .option('escape', '"') \
        .csv(output_path_process)


hdfs_output_path = "hdfs:///output/"+ timestamp
df.write.parquet(hdfs_output_path)
```

https://blog.naver.com/minimine_95

It shows the state of data being put in Hadoop



https://blog.naver.com/minimine_95

Automate pipelines to allow code to proceed automatically whenever new data is created

```python
def run_dataproc(bucket_name, file_name):
    # Dataproc 작업 설정
    project_id = 'db-project-381514'
    region = 'us-central1'
    cluster_name = 'cluster-9bf0'

    job_client = dataproc.JobControllerClient(client_options={"api_endpoint": f"{region}-dataproc.googleapis.com:443"})

    timestamp = str(int(time.time()))  # 타임스탬프 생성
    job = {
        'reference': {
            'project_id': project_id,
            'job_id': timestamp
        },
        'placement': {
            'cluster_name': cluster_name
        },
        'pyspark_job': {
            'main_python_file_uri': 'gs://dbproject_comment/new_gcs.py',
            'args': ['gs://' + bucket_name + '/' + file_name]
        }
    }

    # Dataproc 작업 실행
    response = job_client.submit_job_as_operation(request={"project_id": project_id, "region": region, "job": job})
    print('Dataproc job submitted:', response.operation.name)
```

```python
def gcs_trigger(event, context):
    # 파일 생성 트리거 이벤트 처리
    if event['name'] and event['contentType'] == 'text/csv':
        bucket_name = event['bucket']
        file_name = event['name']

        # 파일이 comment_output 폴더에 저장된 경우에만 실행
        if 'comment_output/' in file_name:
            run_dataproc(bucket_name, file_name)

    return 'Success'
```

https://blog.naver.com/minimine_95

YouTube KR Web process & design

The code to upload YouTube videos

```
<!-- Section -->
    <section>
        <header class="major">
            <h2>Your Video</h2>
        </header>
        <div class="features">
            <iframe width="800" height="500" src="https://www.youtube.com/embed/Os_heh8vPfs" frameborder="0"
            allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen
            style="margin-left: 50px;"></iframe>
        </div>
    </section>
```

https://blog.naver.com/minimine_95

Code that represents csv in HTML table format

```
<table id="content"></table>
<script>
// GCS에서 가져올 파일의 정보
const apiKey = 'AIzaSyAnjzUJVi7wdECQrZVHcZ1DW2C-HYwIUUo';
const bucketName = 'dbproject_comment';
// GCS에서 CSV 파일을 가져와서 테이블에 표시하는 함수
function getFileFromGCS(fileName) {
  const url = `https://storage.googleapis.com/storage/v1/b/${bucketName}/o/${encodeURIComponent(fileName)}?alt=media&key=${apiKey}`;
  $.ajax({
   url: url,
       type: 'GET',
       success: function(data) {
       // CSV 파일을 파싱하여 HTML 테이블로 변환
       const rows = data.split('\n');
       let tableHtml = '<table><thead><tr>';
       const headers = rows[0].split(',');
       for (let i = 0; i < headers.length; i++) {
           tableHtml += '<th>' + headers[i] + '</th>';
       }
       tableHtml += '</tr></thead><tbody>';
       for (let i = 1; i < rows.length; i++) {
           const cells = rows[i].split(',');
           let rowHtml = '<tr>';
           for (let j = 0; j < cells.length; j++) {
           rowHtml += '<td>' + cells[j] + '</td>';
           }
           rowHtml += '</tr>';
           tableHtml += rowHtml;
       }
       tableHtml += '</tbody></table>';

       // HTML 테이블을 페이지에 표시
       $('#content').html(tableHtml);
       },
       error: function(xhr, status, error) {
       console.error('Error fetching file:', error);
       }
  });
}
```

https://blog.naver.com/minimine_95

Code that accesses a folder in the bucket and executes the code on the previous page

```
// GCS에서 버킷의 모든 파일 목록을 가져오는 함수
function listFilesInFolder() {
    const folderName = 'processing_output/';
    const url = `https://www.googleapis.com/storage/v1/b/${bucketName}/o?key=${apiKey}`;
    $.ajax({
        url: url,
        type: 'GET',
        success: function(data) {
            // 'items' 배열에는 버킷의 모든 파일 정보가 들어 있음
            const items = data.items;
            for (let i = 0; i < items.length; i++) {
                const fileName = items[i].name;
                // 파일 이름이 'processing_output/'로 시작하고 '.csv'로 끝나는지 확인
                if (fileName.startsWith(folderName) && fileName.endsWith('.csv')) {
                    // CSV 파일을 처리하는 함수
                    getFileFromGCS(fileName);
                }
            }
        },
        error: function(xhr, status, error) {
            console.error('Error fetching file list:', error);
        }
    });
}

listFilesInFolder();

// update to get a new comment data
setInterval(listFilesInFolder, 5000);
```

https://blog.naver.com/minimine_95

Code for getting spam_output(spam csv) by running only different folder names

```
// GCS에서 버킷의 모든 파일 목록을 가져오는 함수
function listFilesInFolder2() {
    const folderName = 'spam_output/';
    const url = `https://www.googleapis.com/storage/v1/b/${bucketName2}/o?key=${apiKey2}`;

    $.ajax({
        url: url,
        type: 'GET',
        success: function(data) {
            // 'items' 배열에는 버킷의 모든 파일 정보가 들어 있음
            const items = data.items;
            for (let i = 0; i < items.length; i++) {
                const fileName = items[i].name;
                // 파일 이름이 'processing_output/'로 시작하고 '.csv'로 끝나는지 확인
                if (fileName.startsWith(folderName) && fileName.endsWith('.csv')) {
                    // CSV 파일을 처리하는 함수
                    getFileFromGCS2(fileName);
                }
            }
        },
        error: function(xhr, status, error) {
            console.error('Error fetching file list:', error);
        }
    });
}

listFilesInFolder2(); // 페이지 로드시 파일을 불러옴

// 파일을 주기적으로 업데이트
setInterval(listFilesInFolder2, 5000);
```

https://blog.naver.com/minimine_95

YouTube KR

Web process & design

Code for representing a circle with the number of rows in the table

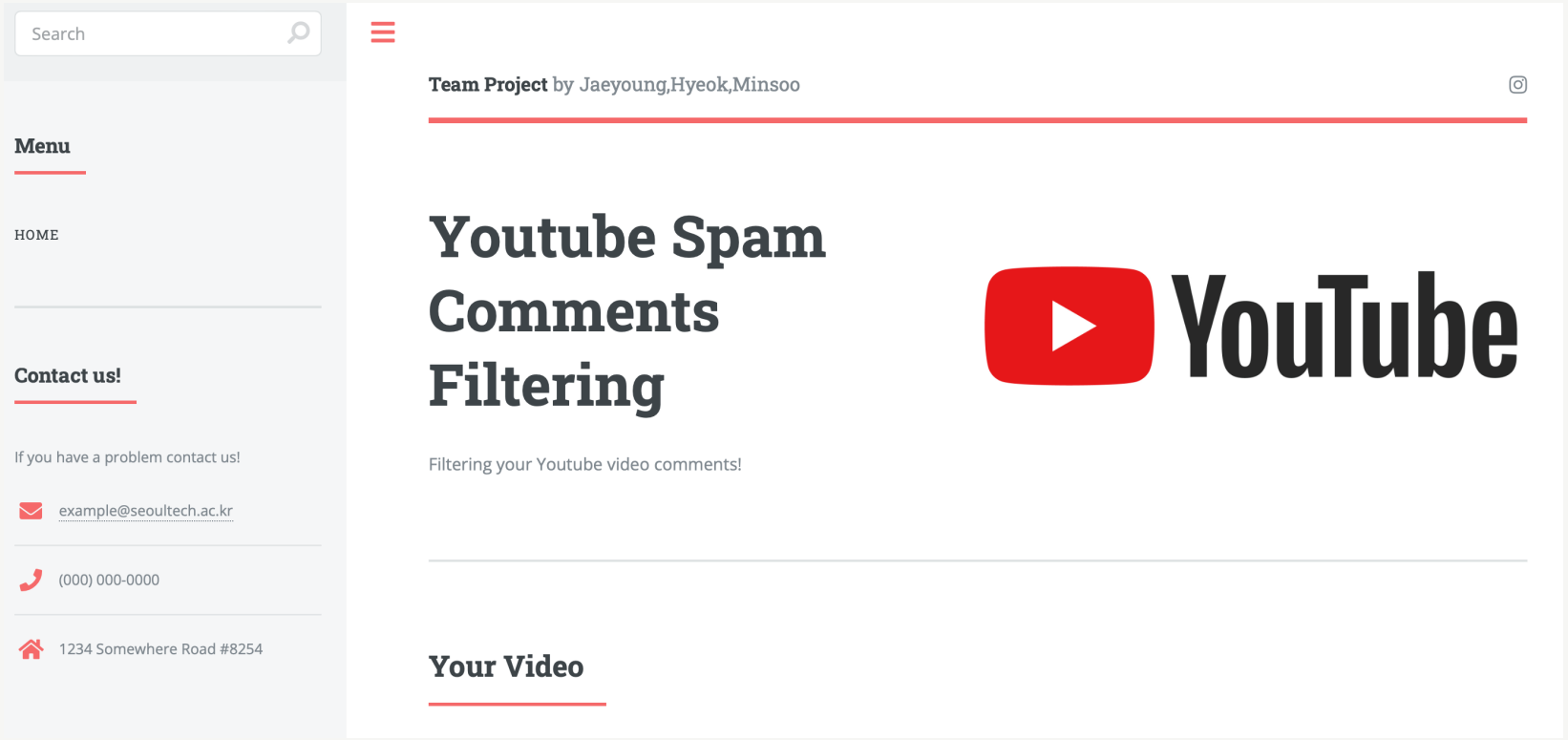
```
</header>
<div class="features">
    <canvas id="myChart" width="800" height="600"></canvas>
</div>
    <script>
        function drawChart() {
            if (myChart) {
                myChart.destroy();
            }
            // var spamData = $('#content2 tr').length;
            var spamData = $('#content2 tbody tr').length-1;
            var notspam = $('#content tbody tr').length -1 - spamData;

            var ctx = document.getElementById('myChart').getContext('2d');
            var myChart = new Chart(ctx, {
                type: 'pie', // 원 그래프를 그리기 위해 'pie'를 설정
                data: {
                    labels: ['Not spam comments', 'Spam Comments'],
                    datasets: [{
                        data: [notspam, spamData],
                        backgroundColor: [
                            'rgba(75, 192, 192, 0.2)',
                            'rgba(255, 99, 132, 0.2)'
                        ],
                        borderColor: [
                            'rgba(75, 192, 192, 1)',
                            'rgba(255, 99, 132, 1)'
                        ],
                        borderWidth: 1
                    }]
                },
                options: {
                    responsive: false // 이것은 차트 크기를 canvas 요소의 크기에 맞추도록 설정
                }
            });
        }
    // 데이터가 변경될 때마다 원 그래프를 업데이트
    setInterval(drawChart, 5000);
```

https://blog.naver.com/minimine_95

YouTube KR

Web process & design

Main web page of our project



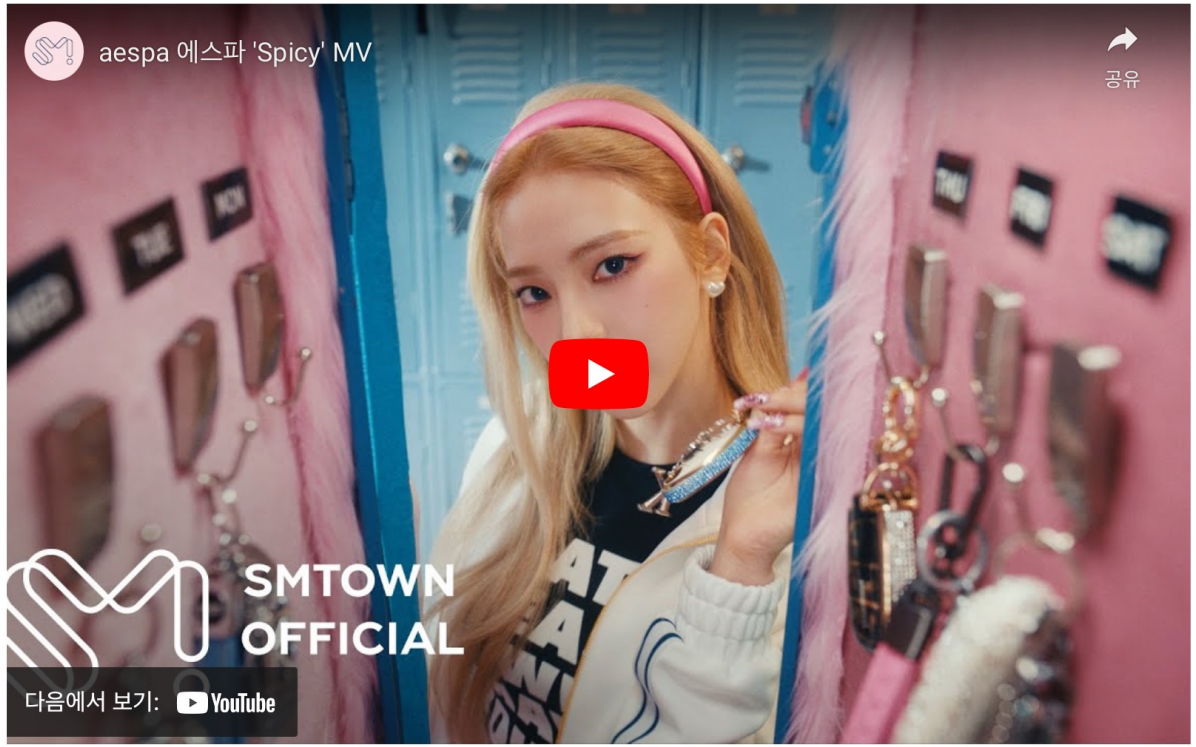
https://blog.naver.com/minimine_95

YouTube KR

Web process & design

Showing videos from YouTube

Your Video



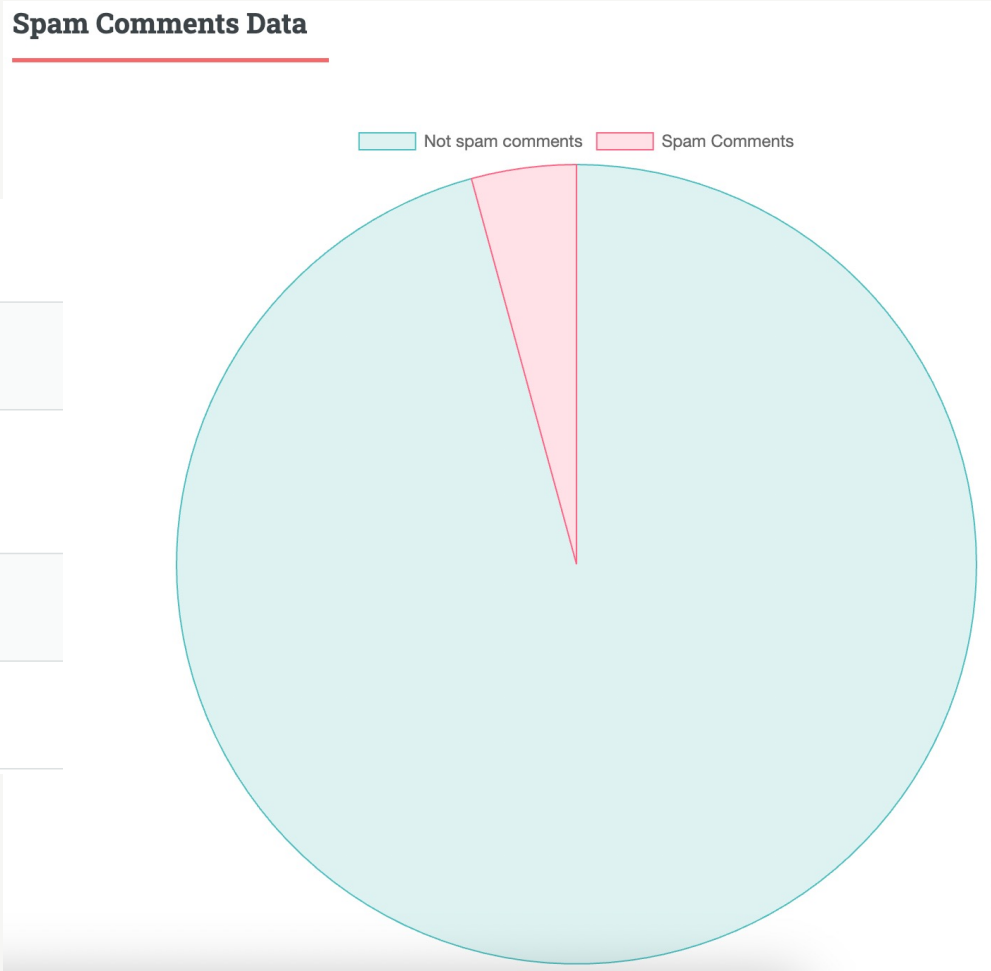
https://blog.naver.com/minimine_95

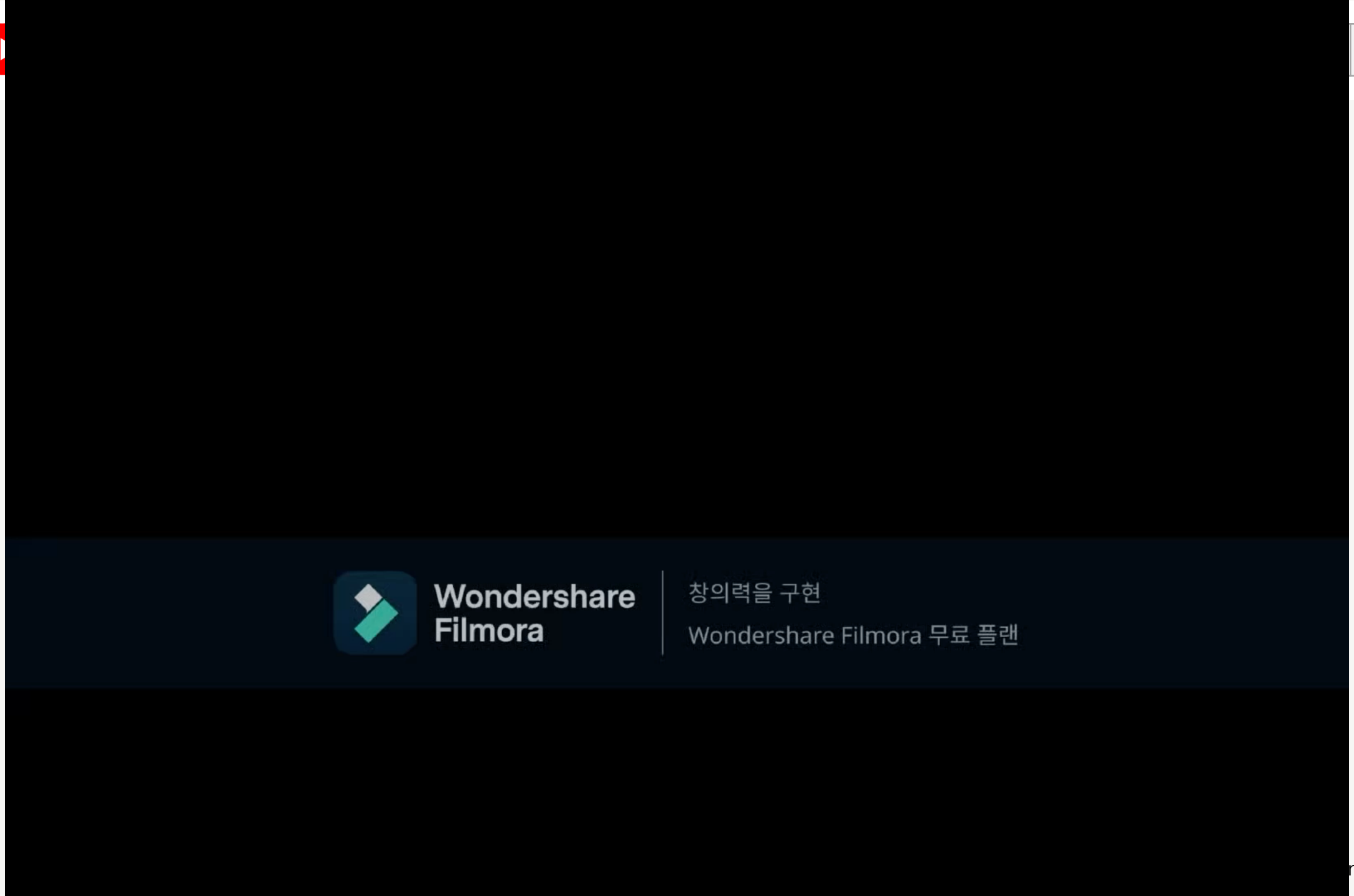
YouTube KR

Web process & design

Showing detected spam comments & Graph of spam comments percentage of all comments

You areawinner U have been specially selectedreceive 1000 or a 4 holiday brspeak to a live operator 2 claim 0871277810910pmin 18	딴딴딴	2023-05-28T15:04:34.000Z	0	1	0.9993709
a hrefhttpswwwyoutubecomwatchvcwLXqtVMbVsampt20m23s2023a	주원	2023-05-10T09:02:58.000Z	0	1	0.9841054
a hrefhttpswwwyoutubecomwatchvcwLXqtVMbVsampt12m22s1222a that bridge is so fucking good I love Ningning39s vocal	Emily von Richter	2023-05-09T14:03:24.000Z	0	1	0.9926588
My ranking br1 I39m Unhappybr2 Welcome to my world br3 Thirstybr4 39Till we meet againbr5 Salty amp Sweetbr6 SpicybrbrYes I LOVE the despressive vibes	TV	2023-05-09T05:41:59.000Z	2	1	0.9582173
My ranking favoritesbr1 Thirstybr2 I39m Unhappybr3 Salty and Sweetbr4 39Til we meet againbr5 Welcome to My Worldbr6 Spicy	P. Marcel	2023-05-08T17:15:07.000Z	5	1	0.9603561





nimine_95

Thank you

https://blog.naver.com/minimine_95