Web Application & Programming Class Project Final Report

**AI, Please let me know the   
air condition!**

Class Name: Web Application Programming

Professor: AICH SATYABRATA

Submission Date: December / 11th / 2024

Name: KIM MINJAE

Student ID: 201924426

**Index**

Content

[**1. Introduction of “What’s up with your city”** 3](#_Toc184822961)

[2. Service Features 5](#_Toc184822962)

[2.1 Implemented Features 5](#_Toc184822963)

[2.1.1 Basic Air Pollution Monitoring 5](#_Toc184822964)

[2.1.2 AI Chatbot based on HuggingFace API 8](#_Toc184822965)

[2.1.3 Prediction of next 7 days 10](#_Toc184822966)

[2.2 Trial & Errors 13](#_Toc184822967)

[2.2.1 Convert tensorflow model to tensorflow JS file to apply on website 13](#_Toc184822968)

[2.2.2 Prompt Engineering details : **SUCCESS** 14](#_Toc184822969)

[2.2.3 gpt2 fine tuning : FAILED 15](#_Toc184822970)

[2.2.4 Training GRU model : **SUCCESS** 16](#_Toc184822971)

[3. APIs & tools Used 17](#_Toc184822972)

[4. Future Directions 18](#_Toc184822973)

# **1. Introduction of “What’s up with your city”**

This project’s goal is not only monitoring the weather and air pollution level of different cities, but also focusing on using AI technologies.

First, To prepare for the future, we need to predict future’s air condition too. It is proper to predict air pollution data to gru model. GRU model is specified to train time series data. Weather data is also time series data. I choosed GRU model. GRU is one of RNN based model which is made to train data that have time dependency.   
Secondly, I choiced chatbot model in huggingface hub which is open AI models & datasets hub. It is difficult to understand air pollution level to normal public. It is not so intuitive to understand that pm2.5 is 10. But, If we ask it to well trained AI model, it reponses us proper answer and gives us some advices.

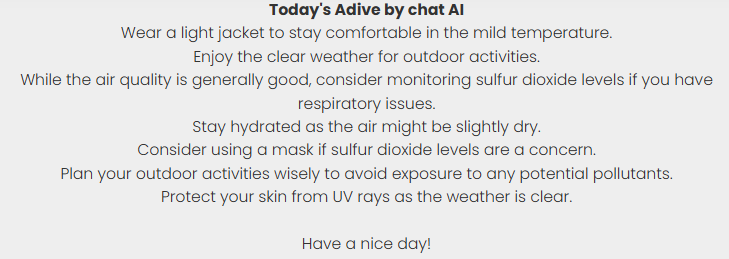
For these reasons, I made web application that gives us basic weather informations and great advices for today’s weather.  
  


Figure : Response By Chatbot

텍스트, 스크린샷, 라인, 도표이(가) 표시된 사진

자동 생성된 설명

Figure : Predicted Air Pollution Level of next 7 days

**Basic view of Website**

텍스트, 스크린샷, 폰트, 평행이(가) 표시된 사진

자동 생성된 설명

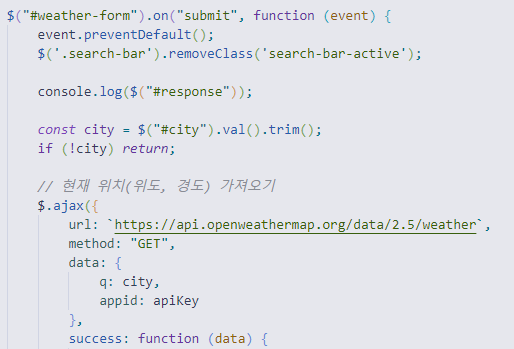
Figure : Web Overview

# 2. Service Features

## 2.1 Implemented Features

### 2.1.1 Basic Air Pollution Monitoring

To show users basic informations about today’s weather and air pollution level, I collected data from **OpenWeatherAPI**. This API service provides data related to weathers like temperature, weather, humidity, and air pollution level. To get data, I signed up and made API key. And to get a data from this, we have to request data using this API key.



Code : base code to request Weather API

I used three APIs to complete this application.  
- Current Weather API: To show today’s temperature, weather, etc..  
- Air Pollution API: To show today’s air pollution level  
- Historical API: To show periodical data(1 year, 1 month)

Using these, First part that I made is show today’s weather and air pollution section. I made it with simple colored box to get to know the level easily.

텍스트, 스크린샷, 직사각형, 도표이(가) 표시된 사진

자동 생성된 설명

Figure : Today's weather & air level with simple color

텍스트, 스크린샷, 직사각형, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure : Another city to comparison

More Red oriented color means bad air conditions, and more green oriented color means better air conditions.

And another section of basic view is plotting the air pollution level.  
There are 7 metrics: co, no, nh3, pm2.5, pm10, o3, so2.  
I request these data for 30 days and compare past data by plotting them.  
To plot them, I used Chart.js.

텍스트, 그래프, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

Figure : plotting the pollution metric

And Today’s pollution value is highlighted by purple line. Other metrics also plotted like this way.  
To show the ratio of them, also plot the pie chart.

텍스트, 스크린샷, 도표, 원이(가) 표시된 사진

자동 생성된 설명

Figure : Pie chart show the ratio of pollution metrics

### 2.1.2 AI Chatbot based on HuggingFace API

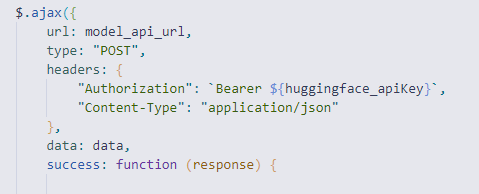
It can be hard to understand each of metrics. And, for the busy people, there’s no time for them to check everything. So I used Chatbot based on gpt3.5. It gives us summary of today’s weather.  
To use this chatbot, I need to register to huggingface which is biggest AI model hub and make APIkey for it.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Figure : Create access token for using chatbot

Among the chatbot, I choosed “Qwen2.5-Coder-32B-Instruct”. Same to request in Weather API part, I also coded request section for it.



Code : Request chatbot using huggingface API

And to get a proper answer by chatbot, I should make proper prompt.  
I gave AI today’s temperature, weather, humidity, pm2.5, ozone, and so2 level. Here is used prompt to get a proper answer.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Code : Prompt for chatbot

As it is mentioned in later section, It is required that make a prompt to get proper answer.  
Answers is limited. So response for AI chatbot isn’t concluded sometimes.

텍스트, 폰트, 스크린샷, 정보이(가) 표시된 사진

자동 생성된 설명

Figure : Incompleted answer

So, we need to process it to get a complete answer.  
First, I trimmed the text and ended text until last dot(.)

### 2.1.3 Prediction of next 7 days

Based on GRU, I made the model predict the future air pollution data based on various cities.  
But before train model with that cities, I should collect historical data from OpenWeather API.  
There are details in “collecting\_air\_pollution\_dataset.ipynb” file in my project.  
I collected data from big cities around the world(Seoul, Busan, Shanghai, Tokyo, New Delhi, London, Paris, New York, California.).  
Because we are living in Busan, I choosed base city to Busan.  
Here is my method to train model.

1. Train baseline model using 3 years of base city(Busan)
2. Fine tune model using other city’s 1 year data.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure : Various cities air pollution datasets

And here is the structure of GRU model.

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

Figure : GRU model structure

And to make model more generalized, I preprocessed data.

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

Figure : loss graph of baseline model

And, now I can predict the future air pollution level for 7 days.

텍스트, 스크린샷, 라인, 도표이(가) 표시된 사진

자동 생성된 설명

Figure : Prediction for 7 days

This model is on the ‘/models/mj\_fine\_tuned\_gru\_model.keras’

## 2.2 Trial & Errors

### 2.2.1 Convert tensorflow model to tensorflow JS file to apply on website: **FAILED**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure : Failed to load tensorflow js converter

I tried various versions of tensorflow including 2.15.0, 2.2.0, etc. I couldn’t find out the reason why It’s not working that convert .h5 or .keras ai model to js code.  
I’m also tried to get my model by uploading it in huggingfaceAPI and using request api.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Figure : gru model uploaded in huggingface

I found tensorflow js. So, I tired to make the same model with node js and tensorflowjs library.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure : Tensorflow JS model

But some of model’s layers are incapable.(like, reshape layer doesn’t work same to python based model) Also, I used more than two preprocessors of scickit-learn. It’s hard to implement the same preprocessors of it.

In the future, maybe I can make tensorflow js model by finding available version of converter or by making tensorflow js model using node js from scratch.

The tensorflow js gru model code is in the project folder.

## 2.2.2 Prompt Engineering details : **SUCCESS**

Normally chatbot use markdown language to show informations. But to apply markdown to my website, it’s hard to convert every markdown grammers. And huggingface API has limited token, It is hard to get a good response. For example, response include the markdown code.  
So I made a prompt like this.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Code : Prompt engineering for proper answer

With this prompt, I made it to get a brief and clear summary and advices for today’s weather.  
And to get rid of useless blank, I used trim() method and ended sentence with last dot(.)



Code : Postprocessing of response

2.2.3 gpt2 fine tuning : FAILED

Before trying to use pretrained chatbot, I tried to use gpt2 model.  
But, not fine-tuned gpt2 only gave us useless corpus.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Figure : Gpt2 can't answer for question

It’s because gpt-2 is the model predict the next word. Not specialized to answer questions.  
So, Based on collected weather data, I tried to fine tune this model.  
But, to train this model, I have to leave some comment for each day’s value.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Figure : Impossible to leave comment to every single data

I dealt with one 3 years data, and 8 1 year data, It is impossible to leave comment to every single data. And It should be inaccurate because I’m not a professional of weather and air condition.

For these reasons, I choosed well-trained chatbot model for give advices for today’s weather.

### 2.2.4 Training GRU model : **SUCCESS**

As I mentioned the former section(2.1.3), I used 9 cities dataset for training.

And the model is optimized to various cities’s datasets.

# 3. APIs & tools Used

**OpenWeatherAPI**: Used to collect weather and air conditions dataset

**HuggingFaceAPI**: Used to load chatbot model

**Chart.js**: Used to visualize the pollution value

**JQuery.js**

**Juypter Notebook**: Used to make a code for collecting data

**Tensorflow**: Used to make a GRU model for prediction for next 7 days air condition

**TensorflowJS**: Used to convert tensorflow ai model to js file

**Scikit-Learn:** Used to preprocess the weather datasets

# 4. Future Directions

With this application, I could not only see the air condition, but also get some proper advices by AI. Even though I failed to convert GRU model to JS file, It still works on python environment. So If I can convert model without any error, This app can be the greate predictor.

I expect to collect more dataset from all around the world and make more precise predictions.  
In addition, I also expect to add more features like pollution map based on pollution metrics.