An AQI Prediction Platform System Design for ABC Company

1. Introduction

1.1 Project Background

Millions of people live in areas where air pollution can cause serious health problems. Individual and corporate users are increasingly concerned about air quality and need a way to obtain air quality predictions to take preventive measures against harm to their health.

1.2 Project Goals

To build a high-precision, scalable global city air quality prediction system that provides accurate and timely AQI prediction information for corporate and individual users, and utilizes generative AI technology to provide personalized air quality information displays for individual users.

1.3 Scope

- Data sources (NOAA GSOD, OpenAQ)
- Prediction time (next 24 hours)
- User groups (corporate users, individual users)
- GenAI image generation

2. Requirements Analysis

2.1 Functional Requirements

2.1.1 Data Acquisition and Processing

The National Oceanic and Atmospheric Administration (NOAA) Global Surface Summary of the Day, dataset URL: <https://registry.opendata.aws/noaa-gsod/>. This data is open data. There are no restrictions on the use of this data. Data documentation address: <http://www.ncdc.noaa.gov/>. This data is stored in AWS S3 and can be obtained directly from S3.

Global aggregated physical air quality data comes from government, research-grade, and other public data sources. Dataset URL: <https://registry.opendata.aws/openaq/>. Data license complies with CC BY 4.0, documentation address: <https://openaq.org>. This data requires registration and obtaining an API key from the official website.

The original data has many problems, with some data having large deviations and some data having serious deficiencies. Before ML, it is necessary to perform EDA on the data. Based on the EDA results, operations such as aggregation, cleaning, and filtering need to be completed through code.

2.1.2 Model Development and Training

The goal of model development is to accurately predict AQI based on weather conditions. GOSD weather data and OpenAQ AQI data are used as input data sources.

Based on the EDA results and feature engineering, only climatic factors that affect AQI are retained. AutoGluon's text model is used for training. AutoGluon can automatically perform data normalization, standardization, and other operations. Model optimization will also be performed automatically. Finally, methods such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) are used to evaluate the training results.

2.1.3 AQI Prediction Service

The trained AutoGluon model is used and deployed to the backend service to predict AQI for the daily acquired weather data, and provide API services for web-side calls.

For individual users' health reminder pictures, the GenAI service provided by Stable Diffusion is used to provide a text-to-image function. Prompt engineering is used to pre-set prompts to guide GenAI to generate pictures. The generated pictures and AQI results are stored together and provided as an API for web-side calls.

APIs for AQI acquisition and health pictures are provided. Different results are returned according to the user type and displayed on the front-end page.

A user login API is provided to distinguish between enterprise users and individual users and return different content.

2.2 Non-Functional Requirements

2.2.1 Performance Requirements

The average response time of the API is less than 100ms. The system supports 1000 concurrent requests per second.

2.2.2 Reliability Requirements

The system availability reaches 99.99%. The data backup frequency is one full backup per week and one incremental backup per day.

2.2.3 Security Requirements

The system adopts a login system to reject direct access to system APIs; implements strict access control to prevent unauthorized access.

2.2.4 Scalability Requirements

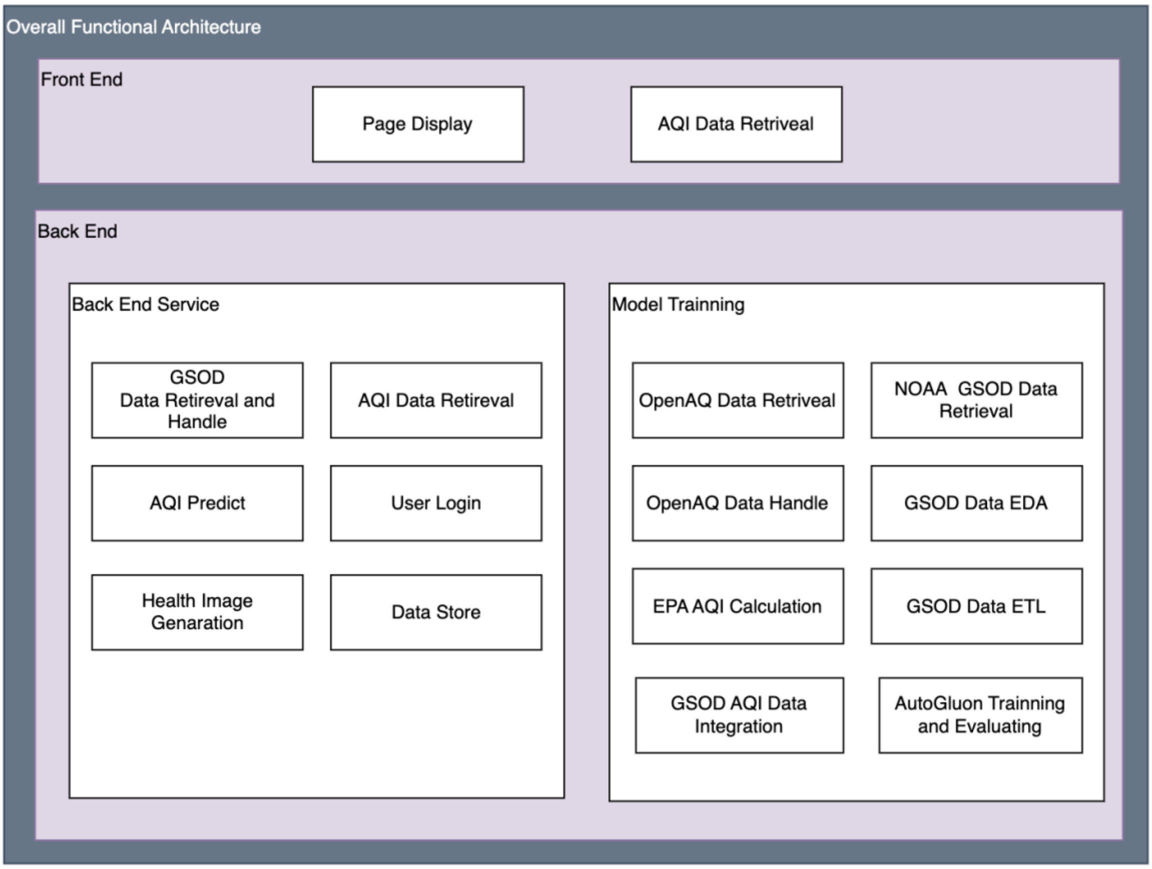
The system supports horizontal scaling. Load balancing software such as Nginx can provide greater concurrency based on the increase in user volume and functions; and provides a multi-level caching system such as CDN, Nginx, Redis, and local cache to increase concurrency.

2.2.5 Cost-Effectiveness

The system uses AWS-related services or components, which allows the company to focus on its business, while Amazon is responsible for management and maintenance, reducing the overall complexity of development and maintenance costs.

3. Solution Design

3.1 Overall Functional Architecture



The system is generally divided into three parts: model training, backend services, and front-end pages. Model training is divided into three main stages: data acquisition, data normalization, and model training evaluation. The backend services include the GSOD data acquisition module; the AQI prediction module based on the model trained by the model training layer; the GenAI-based health picture generation module based on the prediction results and specific city conditions; the user login authentication module; the data storage module; and AQI data and other API response modules. The front end is mainly divided into static pages, login authentication, and AQI acquisition and processing.

3.2 Data Acquisition and Storage Design

3.2.1 Data Source Analysis

NOAA GSOD data is obtained from AWS S3 storage: s3://noaa-gsod-pds/. OpenAQ data is obtained from <https://registry.opendata.aws/openaq/> using the HTTP protocol. GSOD data contains a lot of weather data such as temperature, pressure, wind, precipitation, etc., which needs to be filtered as needed. OpenAQ data contains data on six major pollutants. Some sites only have data on some pollutants, which needs to be aggregated as needed.

3.2.2 Data Storage

Training data storage: Since it is a demo project, the amount of collected data is small, and it is stored in the format of local CSV files. The EDA post-processing results and OpenAQ data aggregation processing results are also stored in local CSV file format.

Actual prediction data storage: Actual prediction data is stored in the MySQL relational database to facilitate API queries and other operations.

3.2.3 Data Processing Flow

First, EDA is performed. Pandas is used to get an overall understanding of the data. Pandas' histograms and box plots are used to understand the distribution of the data. Feature engineering methods are used for specific data analysis. Based on the results of EDA and feature engineering, operations such as aggregation, cleaning, and filtering are performed on the data to obtain integrated data for training and testing inference.

3.2.4 Model Development and Training Design

After analysis and comparison, AutoGluon from AWS is selected for AutoML. Since the AQI value is to be predicted, it is not a binary model, so AutoGluon's regression model is selected.

AutoGluon has many excellent features. It can be trained quickly with only a small amount of code. It uses cloud predictors and pre-built containers to move from experimentation to production. It can be easily extended through custom function processing, models, and metrics.

3.2.5 Training

Use AutoGluon's TabularPredictor for training. Use the prepared training data. Use the TabularPredictor's fit method. Select best_quality for the presets parameter for training. AutoGluon will automatically compare and tune to train the best-performing model.

After training the model, use the trained model to make predictions, and evaluate the prediction ability of the model based on the predicted value and the actual value. Use the MAE, RMSE, and R2 methods for inference evaluation.

3.2.6 Model Deployment

After the AutoGluon model is trained, it is saved as a Predictor object by default. You can directly load it, or encapsulate it into an API, package it into a Docker image, or deploy it to a cloud platform. It is very flexible and convenient.

3.3 AQI Prediction Service Design

The system periodically obtains weather data from GSOD for various cities, and uses the trained AutoGluon model to perform AQI prediction. The predicted results are placed in the database, and the front end provides query and display through the API.

3.4 Personal Health Picture Generation

Personal health pictures are a GenAI text-to-image function. It is necessary to construct a prompt based on the city and weather conditions, and use the API to call the large text-to-image model to generate pictures related to the city and weather. We can use Stable Diffusion deployed on AWS or Replicate's Stable Diffusion to generate them. After generation, they can be stored in the database and returned to the front-end users along with AQI information through the API.

3.5 API Service Design

3.5.1 API Interface

The system adopts a Restful style interface. The main APIs are: the interface for obtaining AQI prediction results and personal health picture information, the interface for obtaining city information, the interface for obtaining only AQI prediction information, and the user login authentication interface.

3.5.2 Authentication and Authorization

The system has two built-in default users, namely enterprise and individual users. JWT technology is used for login authentication. Users are associated with user types, and different information is returned to the front end for display according to enterprise users and individual users.

4. Design Assumptions and Constraints

4.1 Data Assumptions

- The data quality of GSOD and OpenAQ can meet the requirements.
- The update frequency of the GSOD data source can meet the real-time requirements of the prediction.
- The historical data of GSOD and OpenAQ meet the requirements for training the model.
-

4.2 Model Assumptions

- There is a suitable machine learning model that can accurately predict AQI.
- There is a suitable GenAI model that can generate pictures that meet the requirements.
- The model can be trained with reasonable resources and time.

4.3 User Assumptions

- Users can understand the meaning and usage of AQI.

- Users have demand for personalized air quality information display.
- Users can accept the performance and availability of the system.

5. Risk Assessment and Response

5.1 Technical Risks

- The data source is unstable, for example, GSOD weather data cannot be obtained.
- The model prediction accuracy is not high.
- The GenAI model generates pictures of substandard quality.

5.2 Data Risks

- Stored data is lost or damaged.
- Data is leaked or tampered with.
- Data quality issues.
- Data compliance issues.

5.3 Response Strategies

- Unstable data sources are a third-party issue, and clear warnings can be given.
- The problem of low model prediction accuracy can be solved by continuously optimizing and training with more data.
- Different GenAI models can be used to compare the quality of generated pictures.
- Backup strategies are used to prevent data loss and damage.
- Security measures are strengthened to prevent data leakage and tampering.
- Data review is strengthened, and warnings are given for problematic data.
- Close attention is paid to data usage agreements, and solutions are found in a timely manner for compliance issues.

6. References

- NOAA GSOD Dataset Documentation.
- OpenAQ Dataset Documentation.
- EPA AQI Standard Documentation.
- AutoGluon Documentation.
- AWS Official Documentation.
- AQI Sample Project provided by GitHub.