High Level Design (HLD)

Life Expectancy Prediction

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 18/05/2023 | 1.0 | Added Introduction and General description | Sathvik N G |
| 21/05/2023 | 1.1 | Added Design details and abstract | Sathvik N G |
| 22/05/2023 | 1.2 | Created and organized whole document | Sathvik N G |
| 25/05/2023 | 1.3 | Updated Design Details | Sathvik N G |
| 27/05/2023 | 1.4 | Checked and added Performance | Sathvik N G |
| 28/05/2023 | 1.5 | Added Conclusion and References | Sathvik N G |
| 31/05/2023 | 2.0 | Updated diagrams in Design Details | Sathvik N G |

**Table of Content**

Document Version Control ……………………………………………………………. 2

Abstract ………………………………………………………………………………………… 4

Introduction ………………………………………………………………………………….. 5

1. Why this High-Level Document? ……………………………………………. 5

2. Scope …………………………….………………………………………………………. 5

3. Definition ………………………………………………………………………………. 5

General Description ………………………………………………………………………. 6

1. Product Perspective ……………………………………………………….………. 6

2. Problem Statement ………………………………………………………..………. 6

3. Problem Solution ……………………………………………………………………. 6

4. Proposed Methodology …………………………………………………….……. 6

5. Further Improvements ……………………………………………………………. 6

6. Data Required ……………………………………………………...…………………. 6

7. Tools Used ………………………………………………………………………………. 7

8. Constraints …………………………………………………………………..…………. 7

9. Assumptions ……………………………………………………………………………. 7

Design Details ……………………………………………………………..…………………. 8

1. Process Workflow ………………………………………………………..…………. 8

2. Error Handling ………………………………………………………………..………. 9

Performance …………………………………………………………………………….……. 10

1. Reusability ……………………………………………………………..………………. 10

2. Application compatibility ………………………………………..………………. 10

3. Resources Utilization ………………………………………………………….……. 10

4. Deployment ……………………………………………………………………..…..…. 10

Conclusion ………………………………………………………………………….……..……. 11

Reference ………………………………………………………………………………..………. 12

**Abstract**

This project aims to predict life expectancy using machine learning techniques. By analysing various factors such as health status, socioeconomic indicators, and relevant variables, a predictive model will be trained to provide insights into life expectancy. The results can contribute to healthcare planning, policy development, and individual decision-making for a healthier and longer life.

**Introduction**

**1. Why this High-Level Design Document?**

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project

**2. Scope**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

**General Description**

**1. Product Perspective**

The life expectancy prediction project is a machine learning-based model that estimates individuals' life expectancy based on various factors. It provides valuable insights for healthcare, lifestyle decisions, and policy-making. By leveraging machine learning, it offers a predictive tool for assessing and predicting life expectancy.

**2. Problem Statement**

To create the machine learning based solution to predict air quality index based on the parameters.

**3. Problem Solution**

Develop the web application to predict the air quality index and quality of air, which can help citizen to decide whether to go out or not and to alert the citizen in particular area if AQI is relatively high.

**4. Further Improvement**

The project can be extended by using smart IoT devices to make it portable and installing it near cities to track the AQI in real time and to alert people to weak mask if the pollution is high or quality of air is becoming worse.

**5. Data Required**

For training the model we need the data that consist of PM2.5, NO2, CO, SO2, O3 and AQI value of different cities with timestamp.

Data is completely depending upon our problem statement

**6. Tools Used**

* Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn are used to build the whole model.
* PyCharm and Visual Studio Code is used as IDE.
* For visualization of the plots, Matplotlib and Seaborn are used.
* Heroku is used for deployment of the model.
* Front end development is done using Stream lit.
* GitHub is used as version control system.

**7. Constraints**

The AQI prediction website should be user friendly. Different model to be created for different cities.

**8. Assumptions**

The main objective of the project is to ingest live data from IoT devices and AQI monitoring parameters for current air based on the training data used by using Machine Learning. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

**Design Details**

**1. Process Workflow**

For identifying the different types of anomalies, we will use a machine learning model. Below is the process flow diagram.

Diagram

Description automatically generated

Model Training and Evaluation

Diagram

Description automatically generated

Deployment Process

Diagram

Description automatically generated

**2. Error Handling**

Initially we got error when connecting Heroku with GitHub which we solved. We then got error displaying the graphs and multiple text were not solving. All of the above errors have been resolved.

**Performance**

**1. Reusability**

AQI Prediction should be as accurate as possible, so that it will not mislead the user. Best possible model will be used to predict AQI. Since we have used Stream lit and documentation is available, our project follows reusability.

**2. Application compatibility**

Since we are using python and it is compatible with any platform, we follow Application compatibility

**3. Resource utilization**

At the initial stage, we were using high space to create the model. Once the model is created, our system only needs at least of 2GB RAM and 1 GB of storage to run the application smoothly. Whenever user tries to predict the AQI, system uses less than 10% of the processing power.

**4. Deployment**

The code is deployed in GitHub. The whole system is live and is hosted on Heroku.

**Conclusion**

This project proposes the machine learning model for AQI prediction. This model can be used for alerting the citizens when there is air quality high. It will be helpful to reduce disease and problems occur by air pollution. It also helps the government to take an action in that area against air pollution.

**References**

1. <https://docs.streamlit.io/en/stable/>
2. <https://scikit-learn.org/stable/user_guide.html>
3. <https://numpy.org/doc/>
4. <https://seaborn.pydata.org/examples/regression_marginals.html>
5. <https://seaborn.pydata.org/examples/scatterplot_matrix.html>
6. <https://matplotlib.org/>
7. <https://pandas.pydata.org/docs/>