AIR QUALITY ASSESSMENT IN TAMIL NADU

PHASE 2 PROJECT SUBMISSION

INCORPORATING A MACHINE LEARING ALGORITHM TO PREDICTIVE MODEL FOR AIR QUALITY ASSESSMENT IN TAMILNADU

STEPS FOR CREATING A PREDICTIVE MODEL FOR AIR QUALITY ASSESSMENT

1. **Define the Problem**: Clearly define the problem you want to solve. Identify the specific air quality metrics you want to predict (e.g., PM2.5, PM10, AQI) and the geographic areas in Tamil Nadu that you want to assess.

2. **Data Collection**: Collect historical air quality data for Tamil Nadu. This data can typically be obtained from government agencies, environmental organizations, or research institutions. You will also need meteorological data, such as temperature, humidity, wind speed, and wind direction, as they can influence air quality.

3. **Data Preprocessing**: Clean and preprocess the collected data. This includes handling missing values, removing duplicates, and converting data into a suitable format for analysis. Consider normalizing or scaling the data if the features have different scales.

4. **Feature Engineering**: Create relevant features that can enhance the predictive power of your model. Feature engineering may involve creating time-based features (e.g., time of day, day of the week), lag features (e.g., previous day's air quality), and spatial features (e.g., proximity to pollution sources).

5. **Data Splitting**: Split your dataset into training, validation, and test sets. Typically, you might use a 70-80% train, 10-15% validation, and 10-15% test split. This allows you to train, validate, and evaluate your model's performance effectively.

6. **Select a Predictive Model**: Choose an appropriate machine learning or statistical model for air quality prediction. Common models include Linear Regression, Random Forest, Gradient Boosting, Neural Networks, and time series forecasting models like ARIMA or LSTM.

7. **Model Training**: Train your chosen model using the training dataset. Tune hyper parameters to optimize the model's performance. Consider cross-validation to ensure robustness.

8. **Model Evaluation**: Evaluate your model's performance using the validation dataset. Common evaluation metrics for air quality prediction include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R2) score. Adjust your model if the performance is not satisfactory.

9. **Model Testing**: Assess your model's performance on the test dataset to ensure it generalizes well to unseen

data. This step helps you estimate how well your model will perform in real-world scenarios.

10. **Model Deployment**: Once you are satisfied with your model's performance, deploy it to make real-time predictions. Create a user-friendly interface or API to provide air quality forecasts to the public and relevant authorities.

11. **Continuous Monitoring and Maintenance**: Regularly update your model with new data to ensure its predictions remain accurate over time. Monitor the model's performance and retrain it as necessary. Be prepared to adapt to changing environmental conditions and data patterns.

12. **Visualization and Communication**: Visualize the air quality predictions and make them easily accessible to the public and decision-makers. Effective communication of the data and its implications is crucial for public awareness and policy decisions.

13. **Compliance and Regulations**: Ensure that your predictive model complies with any relevant environmental regulations and standards. Collaborate with local environmental agencies for guidance and data sharing.

14. **Feedback Loop**: Establish a feedback loop where users can report discrepancies between predictions and actual air quality. Use this feedback to improve your model and address concerns.

15. **Research and Innovation**: Stay updated with the latest research in air quality modeling, meteorology, and machine learning to incorporate advanced techniques and improve your predictive model continuously.

MACHINE LEARNING MODEL TO IMPROVE THE ACCURACY OF PREDICTIVE MODEL OF AIR QUALITY ASSESSEMENT

Machine Learning is a branch of Artificial Intelligence that aims to provide computers with the ability to learn how to perform specific tasks without being explicitly programmed by a human.

Deep Learning (DL) can be seen as an evolution of ML that uses a structure of multiple layers called Artificial Neural Network (ANN). DL algorithms require less involvement of humans because features are automatically extracted.

.

Classical regression-based algorithms

Regression analysis is used to infer the relation between a dependent variable and a set of independent variables. On the basis of this relation, and using the values of the independent variables, the value of the dependent variable is estimated. Regression helps to predict a continuous value. Next we review classical algorithms to carry out regression.

- **Multiple Linear Regression (MLR)**
- **Auto-Regressive Integrated Moving Average (ARIMA).**

Machine learning regression-based algorithms

- **Support vector regression (SVR).** Support vector machines are mainly used in classification problems.
- **Decision trees (DT).** The aim of this algorithm is to design a model for predicting a quantitative variable from a set of independent variables. The algorithm is based on a recursive partitioning. Trees are composed of decision nodes and leaves. DT regression usually is built by considering the standard deviation reduction to determine how to split a node in two or more branches.
- **Random Forest (RF).** Random Forest is based on the generation of several decision trees. The prediction will be the average of the predictions

provided by the different trees. For the construction of each decision tree, a data sample is selected from the training dataset.

- **K-nearest neighbours regression (KNN).** The *k*-nearest neighbours algorithm is often applied to classification problems although it can be also applied to regression problems.