**Project Name : Air Quality Analysis and prediction in Tamil Nadu**

**Project Description :** Develop a machine learning model predict a Air quality such as So2 ,No2 and RSPM .This prediction make a Air quality in Normal Flow .

**Phase 2 : Innovation**

**Description :** Using a Random Forest algorithm to predict the air Quality and Find a accuracy for a prediction.

Air Quality Analysis and Prediction in Tamilnadu

1.Collect and prepare data :

Gather a dataset of Air quality with their So2,NO2 And RSPM rates. Preprocess the data by cleaning the text, removing stop words, and stemming or lemmatizing the words.

2. Choose a Random Forest Technique :

There are many random forest techniques used to predict the value of Air quality.

3. Train Random Forest model.

Feed the prepared data to the model and allow it to learn the relationships between the features and the target variable (Accuracy of SO2 and NO2).

4. Evaluate the model.

Evaluate the performance of the model on a held-out test set. This will give an idea of how well about the model will generalize to new data.

5. Deploy the model.

Once you are satisfied with the performance of the model, it can deploy it to production. This may involve saving the model to a file or deploying it to a cloud-based platform.

Here is an example of how to train a simple feedforward Machine Learning Algorithms such as Random Forest Technique

Random forest in Python :

Import pandas as pd

From sklearn.model\_selection import train\_test\_split

From sklearn.ensemble import RandomForestRegressor

From sklearn.metrics import mean\_squared\_error

Import matplotlib.pyplot as plt

# Load your dataset

Data = pd.read\_csv(‘cpcb\_dly\_aq\_tamil\_nadu-2014.csv’)

# Split the dataset into features (X) and target (y)

X = data.drop(‘AQI’, axis=1)

Y = data[‘AQI’]

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)s

# Create and train the Random Forest model

Rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

Rf\_model.fit(X\_train, y\_train)

# Make predictions on the test set

Y\_pred = rf\_model.predict(X\_test)

# Calculate the Mean Squared Error (MSE) to evaluate the model

Mse = mean\_squared\_error(y\_test, y\_pred)

Print(f’Mean Squared Error: {mse}’)

# Visualize the feature importances

Feature\_importances = rf\_model.feature\_importances\_

Feature\_names = X.columns

Plt.barh(feature\_names, feature\_importances)

Plt.xlabel(‘Feature Importance’)

Plt.ylabel(‘Feature Name’)

Plt.show()

```

Collect and prepare data

Collect and prepare data

Gather a dataset of movies with their IMDb scores. Preproces or lemmatizing the words.

2. Choose a neural network architecture.

There are many different neural network architectures that can be used for regression tasks, such as predicting IMDb scores. Some popular architectures include feedforward neural networks, convolutional neural networks, and recurrent neural networks.

3. Train neural network model.

Feed the prepared data to the model and allow it to learn the relationships between the features and the target variable (IMDb score).

4. Evaluate the model.

Evaluate the performance of the model on a held-out test set. This will give an idea of how well about the model will generalize to new data.

5. Deploy the model.

Once you are satisfied with the performance of the model, it can deploy it to production. This may involve saving the model to a file or deploying it to a cloud-based platform. a dataset of movies with their IMDb scores. Preprocess the data by cleaning the text, removing stop words, and stemming or lemmatizing the words.

2. Choose a neural network architecture.

There are many different neural network architectures that can be used for regression tasks, such as predicting IMDb scores. Some popular architectures include feedforward neural networks, convolutional neural networks, and recurrent neural networks.

3. Train neural network model.

Feed the prepared data to the model and allow it to learn the relationships between the features and the target variable (IMDb score).

4. Evaluate the model.

Evaluate the performance of the model on a held-out test set. This will give an idea of how well about the model will generalize to new data.

5. Deploy the model.

Once you are satisfied with the performance of the model, it can deploy it to production. This may involve saving the model to a file or deploying it to a cloud-based platform.