Phase 2: Innovation

Consider incorporating machine learning algorithms to improve the accuracy of the predictive model.

1. **Data Collection & Pre-processing**:
   - Ensure you have a good quality dataset: Remove outliers, handle missing values, and scale/normalize your data.
   - Split your data into training, validation, and test sets to evaluate the performance of your models.
2. **Feature Engineering**:
   - Create new features based on domain knowledge.
   - Use techniques like Principal Component Analysis (PCA) for dimensionality reduction.
3. **Choose Suitable Algorithms**:
   - For regression problems, you might consider algorithms like Linear Regression, Decision Trees, Random Forest, Gradient Boosting Machines, Neural Networks, etc.
   - For classification problems, Logistic Regression, Support Vector Machines, Naive Bayes, k-Nearest Neighbours, Neural Networks, etc., can be considered.
4. **Model Training**:
   - Use the training set to train various machine learning models.
   - Regularize models (e.g., L1, L2 regularization) to prevent over fitting.
5. **Model Evaluation**:
   - Use the validation set to tune hyper parameters using techniques like Grid Search or Random Search.
   - Evaluate models using appropriate metrics (accuracy, F1 score, ROC curve, MSE, etc.), depending on the problem type.
6. **Model Ensemble**:
   - Combine predictions from multiple models to achieve better accuracy. Techniques like Bagging, Boosting, and Stacking can be employed.
7. **Deploy & Monitor**:
   - Once satisfied with the model's performance on the test set, deploy the model.
   - Continuously monitor the model's performance in real-world scenarios. Re-train the model if its performance degrades.
8. **Iterate**:
   - Continually collect new data, retrain models, and adjust features or algorithms based on new insights or changes in the data distribution.

Program:

```
import pandas as pd

from sklearn.ensemble import RandomForestRegressor

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error

from sklearn.preprocessing import LabelEncoder

data = pd.read_csv("cpcb_dly_aq_tamil_nadu-2014.csv")

print(f"After loading: {data.shape}")

data['Sampling Date'] = pd.to_datetime(data['Sampling Date'], format='%d-%m-%y', errors='coerce')

print(f"After date processing: {data.shape}")

for col in data.columns:

    if data[col].dtype == 'object':

        data[col].fillna(data[col].mode()[0], inplace=True)

    else:

        data[col].fillna(data[col].mean(), inplace=True)

print(f"After handling missing values: {data.shape}")

print(f"Number of NaN values in 'RSPM/PM10': {data['RSPM/PM10'].isna().sum()}")

print(f"Number of NaN values in 'PM 2.5': {data['RSPM/PM10'].isna().sum()}")

if data['PM 2.5'].isna().sum() != len(data):

    data = data[data['RSPM/PM10'].notna()]

    print(f"After filtering PM 2.5: {data.shape}")

else:

    print("The 'PM 2.5' column is entirely NaN. Choose a different target or source the missing data.")

    exit()

data['year'] = data['Sampling Date'].dt.year

data['month'] = data['Sampling Date'].dt.month
```

```
data['day'] = data['Sampling Date'].dt.day
data = data.drop('Sampling Date', axis=1)
label_encoders = {}
for column in ["State", "City/Town/Village/Area", "Location of Monitoring Station", "Agency", "Type of Location"]:
    le = LabelEncoder()
    data[column] = le.fit_transform(data[column])
    label_encoders[column] = le
print(f"After encoding: {data.shape}")
X = data.drop(["RSPM/PM10", "Stn Code"], axis=1)
y = data["RSPM/PM10"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
```

Output:

```
After loading: (2879, 11)
After date processing: (2879, 11)
After handling missing values: (2879, 11)
Number of NaN values in 'RSPM/PM10': 0
Number of NaN values in 'PM 2.5': 0
The 'PM 2.5' column is entirely NaN. Choose a different target or source the missing data.
```

Row Labels	Sum of Stn Code
Industrial Area	262636
Residential, Rural and other Areas	1107049
Grand Total	1369685

Row Labels	Sum of Stn Code
Chennai	408522
Coimbatore	83748
Cuddalore	224961
Madurai	90259
Mettur	156312
Salem	40479
Thoothukudi	82448
Trichy	282956
Grand Total	1369685

SO2	NO2	RSPM/PM10
11	17	55
13	17	45
12	18	50
15	16	46
13	14	42
14	18	43
12	17	51
13	16	46
10	19	50
15	14	48
14	16	32
14	14	29
13	17	17
15	16	44
12	17	25
13	16	29
11	18	29
15	16	41
14	17	43
14	14	42
14	17	54
15	19	62
14	15	66