Install required package

This ensures all libraries needed for data handling, modeling, and plotting are available.

```
install.packages(c("readr", "dplyr", "neuralnet", "xgboost", "ggplot2"))
```

#Load libraries

These libraries handle reading data, preprocessing, training models, and visualization.

```
library(readr)

library(dplyr)

library(neuralnet)

library(xgboost)

library(ggplot2)
```

Load Dataset

Import the air quality dataset from CSV file into R.

```
data <- read_csv("C:/Users/DELL/Downloads/air_quality_prediction_dataset.csv",
show_col_types = FALSE)

if("Date" %in% names(data)) data <- data %>% select(-Date)
```

Remove non-numeric Date column and ensure all other columns are numeric for modeling.

3.Data Cleaning and Preprocessing:

Data preprocessing ensures the dataset is ready for analysis.

Steps performed:

1.Rename columns to avoid dots

```
names(data) <-
c("PM25","PM10","NO2","CO","SO2","O3","Temperature","Humidity","WindSp
eed","AQI")
```

2.Ensure all columns are numeric

```
data[] <- lapply(data, as.numeric)
```

3.Prepare Data for Modeling

Separate features (X) and target variable (Y = AQI)

```
x <- as.matrix(data %>% select(-AQI))

y <- data$AQI
```

4.Train Neural Network

Build a simple neural network model with small hidden layers due to tiny dataset.

```
nn_formula <- AQI ~ PM25 + PM10 + NO2 + CO + SO2 + O3 + Temperature +
Humidity + WindSpeed
```

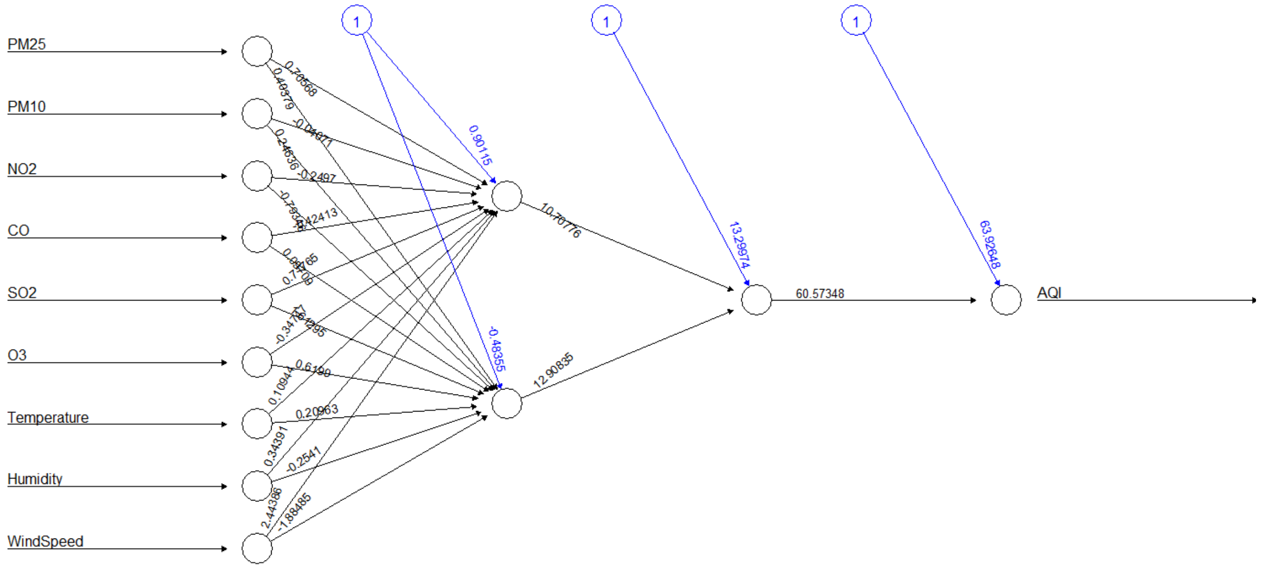
5.Small network for tiny dataset

```
nn_model <- neuralnet(nn_formula, data = data, hidden = c(2,1), linear.output =
TRUE)
```

6.Plot Neural Network

```
plot(nn_model)
```

OUTPUT:



7.Predictions

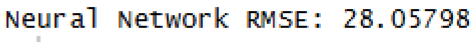
nn_pred <- compute(nn_model, data %>% select(-AQI))$net.result

8.Evaluation

nn_rmse <- sqrt(mean((y - nn_pred)^2))

cat("Neural Network RMSE:", nn_rmse, "\n")

OUTPUT:



```
Neural Network RMSE: 28.05798
```

9.Train XGBoost

Train a boosting-based regression model on the same dataset for comparison.

dall <- xgb.DMatrix(data = x, label = y)

xgb_model <- xgboost(data = dall,

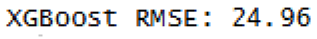
nrounds = 20,   # fewer rounds for small dataset

```
                    objective = "reg:squarederror",
                    max_depth = 2,      # simpler tree depth
                    eta = 0.1,
                    verbose = 0)
```

10.Predictions

```
xgb_pred <- predict(xgb_model, dall)
```

OUTPUT:



```
XGBoost RMSE: 24.96
```

11.Evaluation

```
xgb_rmse <- sqrt(mean((y - xgb_pred)^2))
cat("XGBoost RMSE:", xgb_rmse, "\n")
```

12.Linear Regression Baseline

Train a simple linear regression model as a baseline reference.

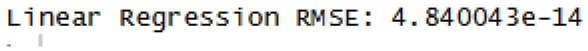
```
lm_model <- lm(AQI ~ PM25 + PM10 + NO2 + CO + SO2 + O3 + Temperature +
Humidity + WindSpeed, data = data)

lm_pred <- predict(lm_model, newdata = data)
```

13.Evaluation

```
lm_rmse <- sqrt(mean((y - lm_pred)^2))
cat("Linear Regression RMSE:", lm_rmse, "\n")
```

OUTPUT:



```
Linear Regression RMSE: 4.840043e-14
```

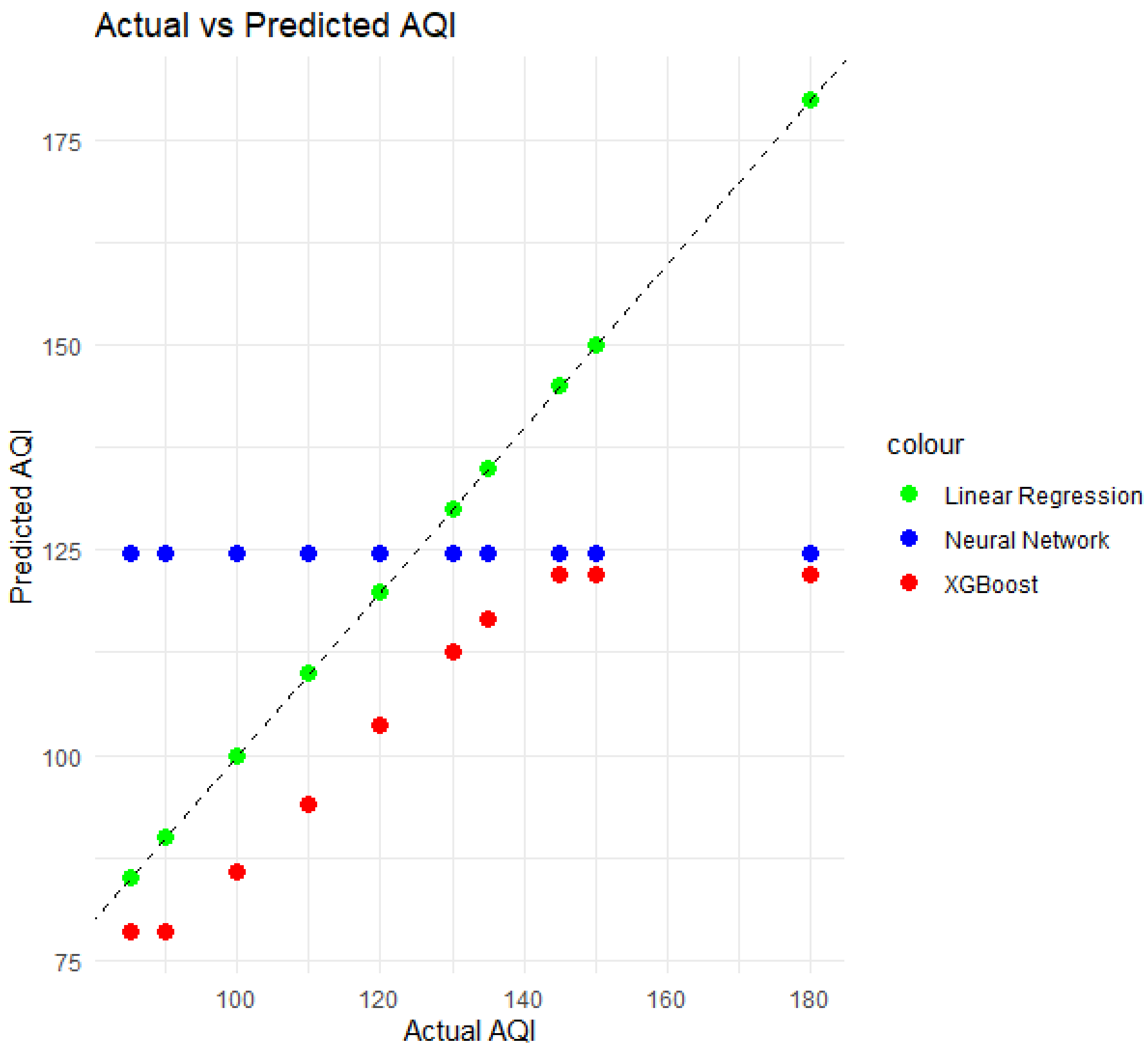
14.Plot Actual vs Predicted for all models

Visualize how well each model’s predictions align with actual AQI values.

```
results <- data.frame(
  Actual = y,
  NN_Predicted = nn_pred,
  XGB_Predicted = xgb_pred,
  LM_Predicted = lm_pred
)


ggplot(results, aes(x = Actual)) +
  geom_point(aes(y = NN_Predicted, color = "Neural Network"), size = 3) +
  geom_point(aes(y = XGB_Predicted, color = "XGBoost"), size = 3) +
  geom_point(aes(y = LM_Predicted, color = "Linear Regression"), size = 3) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  labs(title = "Actual vs Predicted AQI",
       y = "Predicted AQI", x = "Actual AQI") +
  scale_color_manual(values = c("Neural Network"="blue", "XGBoost"="red",
"Linear Regression"="green")) +
  theme_minimal()
```

OUTPUT:



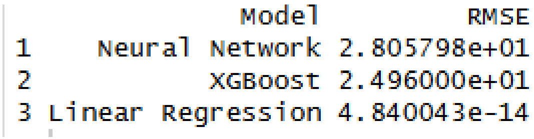
15.RMSE Comparison Table

Compare the model performances in terms of Root Mean Square Error (RMSE).

```
rmse_table <- data.frame(
  Model = c("Neural Network", "XGBoost", "Linear Regression"),
  RMSE = c(nn_rmse, xgb_rmse, lm_rmse)
)
```

print(rmse_table)

OUTPUT:



```
             Model         RMSE
1    Neural Network 2.805798e+01
2           XGBoost 2.496000e+01
3 Linear Regression 4.840043e-14
```