**Data Experiment, Presentation and Analysis Plan Musah**

1. **Load Dataset**: Start by loading the dataset you'll be working with.
2. **Encode Categorical Columns**: Convert any string columns to numeric using encoding techniques if necessary.
3. **Create DataFrame (DF)**: Create a DataFrame to work with.
4. **Split Data**: Split the data into training, testing, and validation sets, with the dependent variable (y) for prediction.
5. **Show Parameters and Frequency Analysis**: Show parameters with units and explore correlations between parameters if possible. Include a clear emphasis on the frequency parameter, particularly for various frequency bands like 1G, 2G, 3G, 4G, 5G, and mmWave.
6. **Model Training History**: During model training, keep track of metrics like MSE, MAE, R2, standard deviation, and corrections.
7. **Create Second Stage DataFrame**: Create a new DataFrame containing all features (X), target variable (y), and predictions.
8. **Model Architecture Determination**: Determine the architecture of the model, including hidden layers, number of neurons, transfer function, learning rate, and the algorithm used. This may involve experimentation and optimization.
9. **Learning Algorithm and Training**: Show a table of learning algorithms, weight adaptation methods, and training acronyms used for training, testing, and validation, along with evaluation metrics like MAE, R2, std, RMSE, coe%, and MSE. Include classical machine learning model names and acronyms like MLR (Multiple Linear Regression), PR (Polynomial Regression), SVR (Support Vector Regression), DT (Decision Trees), RF (Random Forests), KNN (K-Nearest Neighbors), ANN (Artificial Neural Networks), RNN (Recurrent Neural Networks), and LSTM (Long Short-Term Memory).
10. **Define Traditional Empirical Models**: Define functions for traditional empirical models like Walficsh–Ikegami, Walficsh Bertoni, cluster factor, COST 234 Hata, Hata Okumura, SUI model, Lee model, and Egli Model.
11. **Create Third Stage DataFrame**: Create a DataFrame with all features (X), target variable (y), and columns for predictions from the traditional empirical models.
12. **Build Classical Machine Learning Models**: Build classical machine learning models like MLR, PR, SVR, DT, RF, KNN, ANN, and RNN based on LSTM.
13. **Model Training Parameters**: Show a table of all parameters used for training the model for prediction.
14. **Model Performance**: Show a table with the number of neurons and training/testing metrics like MSE, MAE, R2, std, RMSE, and coe%.
15. **Overall Performance Metrics**: Show a table of overall performance metrics like MAE, R2, std, RMSE, and coe%.
16. **Bar Graph Comparison**: Show a bar graph comparing metrics like MSE, MAE, etc., for different models.
17. **Overall Performance Comparison**: Show a table of overall performance metrics for different models.
18. **Path Loss Prediction Comparison**: Show a bar graph comparing path loss prediction for different models.
19. **Performance Graph Comparison**: Show performance graphs comparing error statistics against neurons for different models.
20. **Bar Graph Comparison for Classical ML Models**: Show a bar graph comparing path loss prediction for classical ML models and the trained model.
21. **Performance Graph Comparison for Classical ML Models**: Show performance graphs comparing error statistics against neurons for classical ML models and the trained model.
22. **Performance Graph for Traditional Models**: Show performance graphs comparing error statistics for traditional empirical models and the trained model.
23. **Performance Graph for Classical ML Models**: Show performance graphs comparing error statistics for classical ML models and the trained model.
24. **Transfer Function Table**: Show a table for the transfer function used for training, testing, and validation along with evaluation metrics.
25. **Graph for Standard Deviation**: Show a graph comparing standard deviation against training algorithms/transfer functions.
26. **Graph for RMSE**: Show a graph comparing RMSE against training algorithms/transfer functions.
27. **Graph for R2 Value**: Show a graph comparing R2 value against training algorithms/transfer functions.
28. **Hidden Layer and Neuron Table**: Show a table comparing the number of hidden layers for each algorithm used for training, testing, and validation along with evaluation metrics.
29. **Hyperparameter Turning Table**: Show a table comparing learning algorithms and hyperparameter tuning methods used for training, testing, and validation along with evaluation metrics.
30. **Loss Plot Comparison**: Show a graph representing the comparison of loss plots of path loss against predicted path loss for the trained model.
31. **Correlation Plot**: Show a graph plot of predicted path loss performance correlation coefficient with actual path loss for the trained model.
32. **MSE with Epoch Plot**: Show a graph plot of MSE with Epoch for the trained model.
33. **Mean Prediction Error Graph**: Show a bar graph of mean prediction error against data point statistics for the trained model.
34. **MAE Comparison Bar Graph**: Show a bar graph of MAE representation for traditional empirical models and the trained model.
35. **RMSE Comparison Bar Graph**: Show a bar graph of RMSE representation for traditional empirical models and the trained model.
36. **Standard Deviation Comparison Bar Graph**: Show a bar graph of standard deviation error representation for traditional empirical models and the trained model.
37. **R2 Statistics Comparison**: Show a bar graph of R2 statistics representation for traditional empirical models and the trained model.
38. **Path Loss Line Plot**: Show a graph line plot of path loss for the trained model at different frequencies.
39. **Path Loss Dot Plot**: Show a graph dot plot of path loss for the trained model at different frequencies.
40. **Predicted Path Loss Line Plot**: Show a graph line plot of predicted path loss for the trained model at different frequencies.
41. **Predicted Path Loss Dot Plot**: Show a graph dot plot of predicted path loss for the trained model at different frequencies.
42. **Run Time Table**: Show a table for runtime (seconds) for the trained model along with evaluation metrics.
43. **Environmental Characteristics Table**: Show a table for environmental characteristics (Rural, Urban, Suburban) for the trained model and path loss characteristics (LOS, NLOS).
44. **Environmental Characteristics Graph**: Show a graph plot of predicted path loss, actual path loss representing environmental characteristics for path loss characteristics (LOS, NLOS) at different frequencies for the trained model.
45. **Path Loss Frequency Characteristics**: Plot original path loss against distance for all frequencies, Visualize predicted path loss against each model's path loss. Plot original path loss against distance for all frequencies, Plot predicted path loss against distance. Create a standalone legend for frequency with function assigned to frequency to frequency band.