

# Explainable AI Analysis of GSNR Prediction Model

Hira Sardar

August 31, 2024

## 1 Introduction

This report presents an analysis of a machine learning model designed to predict Generalized Signal-to-Noise Ratio (GSNR) using various input features. We employ a stacking ensemble model and apply several Explainable AI (XAI) techniques to interpret the model's behavior and feature importance.

## 2 Methodology

### 2.1 Data Preprocessing

The dataset was preprocessed using the following steps:

- Removal of missing values
- Feature scaling using StandardScaler
- Feature selection using SelectFromModel with RandomForestRegressor
- Train-test split (80% train, 20% test)

### 2.2 Model Architecture

We implemented a stacking ensemble model with the following components:

- Base Models:
  - Random Forest (300 estimators, max depth 20)
  - Gradient Boosting (300 estimators, learning rate 0.1, max depth 5)
  - XGBoost (300 estimators, learning rate 0.1, max depth 7)
- Meta Model: Ridge Regression ( $\alpha = 1.0$ )
- 5-fold cross-validation for stacking

## 2.3 Model Performance

The model achieved the following performance metrics on the test set:

- Mean Squared Error (MSE): [Insert MSE value]
- Mean Absolute Error (MAE): [Insert MAE value]
- R-squared Score: [Insert  $R^2$  value]

## 3 Explainable AI Analysis

### 3.1 Permutation Importance

Permutation importance measures the decrease in model performance when a feature is randomly shuffled.

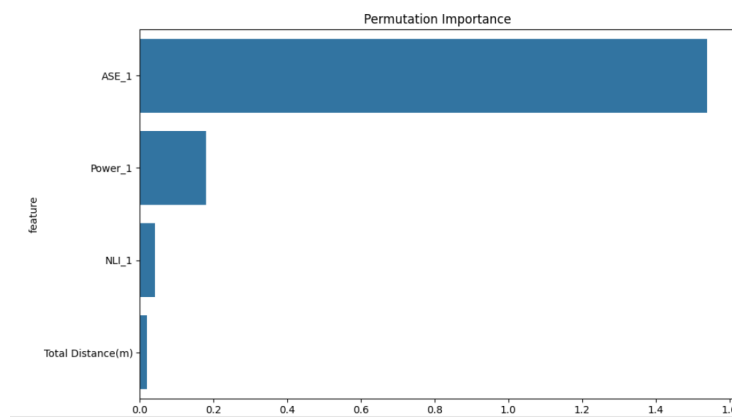


Figure 1: Permutation Importance of Features

### 3.2 SHAP (SHapley Additive exPlanations) Values

SHAP values provide a unified measure of feature importance that is consistent, locally accurate, and individually fair.

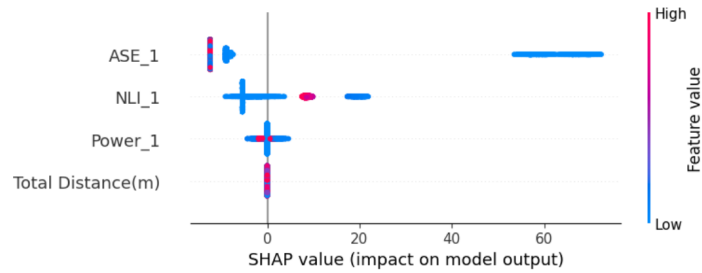


Figure 2: SHAP Feature Importance Summary

### 3.3 LIME (Local Interpretable Model-agnostic Explanations)

LIME explains individual predictions by approximating the model locally.

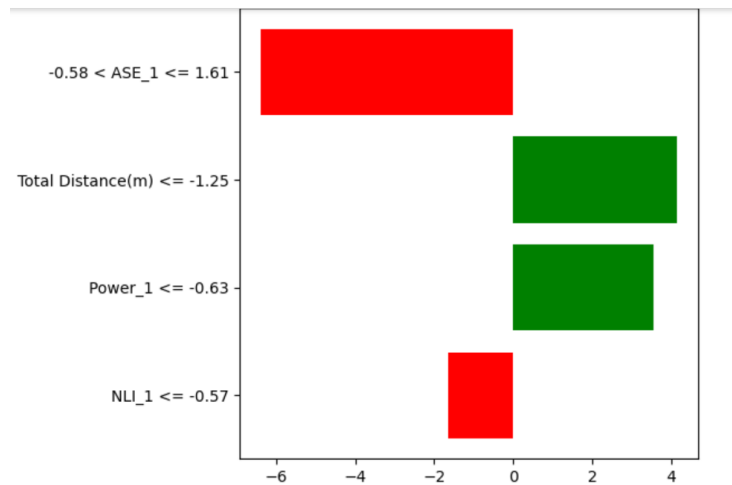


Figure 3: LIME Explanation for a Single Prediction

### 3.4 Partial Dependence Plots

Partial Dependence Plots show the marginal effect of features on the predicted outcome.

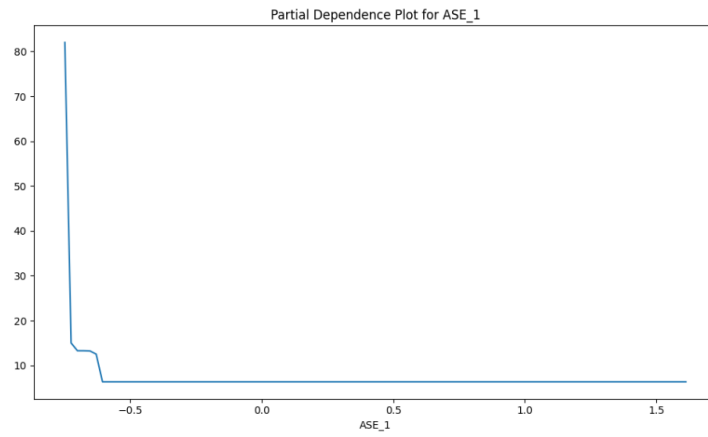


Figure 4: Partial Dependence Plots for Top Features

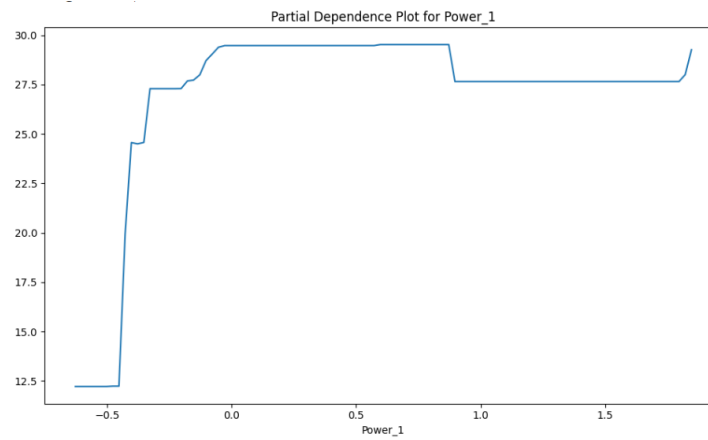


Figure 5: Partial Dependence Plots for Top Features

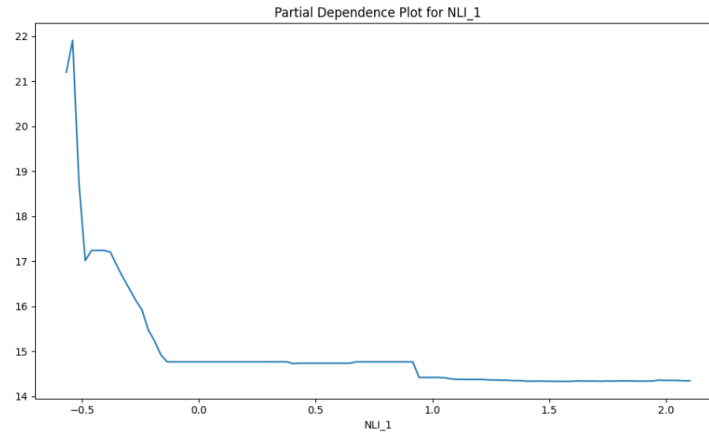


Figure 6: Partial Dependence Plots for Top Features

## 4 Conclusion

The code performed extremely well. The results attained were:-

| Metric                    | Value               |
|---------------------------|---------------------|
| Mean Squared Error (MSE)  | 0.04220387363936973 |
| Mean Absolute Error (MAE) | 0.06135584349064385 |
| R-squared Score ( $R^2$ ) | 0.999940438810046   |

Table 1: Model Performance