📘 Machine Learning Model Report

1.Objective

The goal of this project is to build and evaluate multiple machine learning models to predict “device malfunction” using the given dataset. Emphasis is placed on achieving a balanced trade-off between “Recall” and “F1-score”, especially for the minority class (malfunction = 1).

---

2. 📁 Dataset Overview

- “Filename”: `dataset.csv`

- “Target Variable”: `malfunction` (binary)

- “Key Features”: - “Preprocessing”:

  - Removed `date` and `product` columns

  - Scaled numeric features using `StandardScaler`

  - Balanced the target classes using “SMOTE”

---

3. ⚙️ Modeling Pipeline

✅ Steps:

1. “Train-test split” with stratification (80/20)

2. “Feature scaling” using `StandardScaler`

3. “SMOTE” applied to handle class imbalance

4. Models trained:

  - Logistic Regression

  - Decision Tree

  - Random Forest

  - XGBoost (with `scale_pos_weight`)

📊 Model Evaluation Approach

- For each model:

  - Predicted “probability scores”

  - Used “`precision_recall_curve`” to analyze thresholds

  - Selected the threshold with the “highest F1-score”

- Metrics evaluated:

  - “Precision”

  - “Recall”

  - “F1-score”

  - “Confusion Matrix”

📈 Best Threshold Selection Logic

```python

precisions, recalls, thresholds = precision_recall_curve(y_test, probs) to get the best Threshold for the following Models

## Results :

| Model | Class | Precision | Recall | F1 Score |
|---|---|---|---|---|
| **Logistic Regression** | 0--Non Malfunction | **0.999** | **1** | **0.9994** |
| | 1-Malfunction | **0.188** | **0.143** | **0.1622** |
| **Decision Tree** | 0--Non Malfunction | 0.999 | 0.999 | 0.999 |
| | 1-Malfunction | 0.088 | 0.143 | 0.1091 |
| **Random Forest** | 0--Non Malfunction | 0.999 | 1 | 0.9995 |
| | 1-Malfunction | 0.25 | 0.095 | 0.1379 |
| **XGBoost** | 0--Non Malfunction | 0.999 | 0.998 | 0.9988 |
| | 1-Malfunction | 0.065 | 0.143 | 0.0896 |

The above the following score's : Use **Logistic Regression Model** — it gives the best **F1-score** for the minority class while maintaining a strong overall model with the

"Precision - 0.99 and Recall – 1.00" for Non Malfunction Class" and

"Precision- 0.18 and Recall – 0.143" For Malfunction".

Thank you !!
```