

# System Threat Forecaster

---

HITESH

22F2001256

# Introduction

## Objective

Predict whether a system will be infected by malware using detailed telemetry data collected from antivirus software—including hardware, OS, and configuration settings.

## Importance:

As cybersecurity threats become more frequent and advanced, proactive ML-based detection empowers organizations to identify risks early, strengthen defenses, and minimize system damage.

Notebook Link - [System Threat Forecaster](#)

Contest Link - [Contest](#)

# Background

- **System Telemetry:** Encompasses 76 features such as machine IDs, antivirus product configurations, processor details, OS versions, protection statuses, and geographic identifiers.
- **Task:** The classification task is binary: predict “0” or “1” status for each system.  
(0 – Infected by Malware, 1 – Not infected)
- **Accuracy score** is the chosen metric, reflecting the proportion of correct infection status predictions.
- Key ML processes include data cleaning (handling missing values and outliers), feature encoding, normalization, model selection, and evaluation.

# Data Preprocessing

- **Loading Data:** Data loaded includes training and test CSV files with 100,000 rows and 76 system-related features such as system hardware, OS details, antivirus configs, and geo-location.
- **Cleaning:** Infinite values replaced with NaN and missing values handled by imputation. Some features had less than 1% missing data.
- **Outlier Treatment:** Applied interquartile range (IQR) method to detect and cap outliers to reduce their impact on model learning.
- **Feature Engineering:**
  - Categorical features were imputed using *most\_frequent value* and encoded using Ordinal Encoding for consistent numeric input to models.
  - Standard Scaling applied after full transformation.
- **Pipeline:** Combined preprocessing steps into a scikit-learn Pipeline for smooth integration into model training.

# Exploratory Data Analysis (EDA) Insights

- **General Stats:** Dataset is balanced with approximately 50.5% infected systems.
- **Key Insights:**
  - Systems with more antivirus products installed have a lower infection rate, indicating higher security.
  - Infection rates vary widely by geographic regions; top countries like IDs 29, 43, 141 reported high infection counts.
- **Visuals:** Count plots demonstrated distributions of antivirus product counts against infection.
- Most of the features don't affect the result. All the features are mostly balanced.

# Model Training and Comparison

## 1. Logistic Regression:

- Simple linear model, good baseline but limited in capturing complex interactions.
- Hyperparameter tuning was applied on Logistic Regression for improved accuracy.
- Accuracy Score ~ 59%

## 2. Decision Tree:

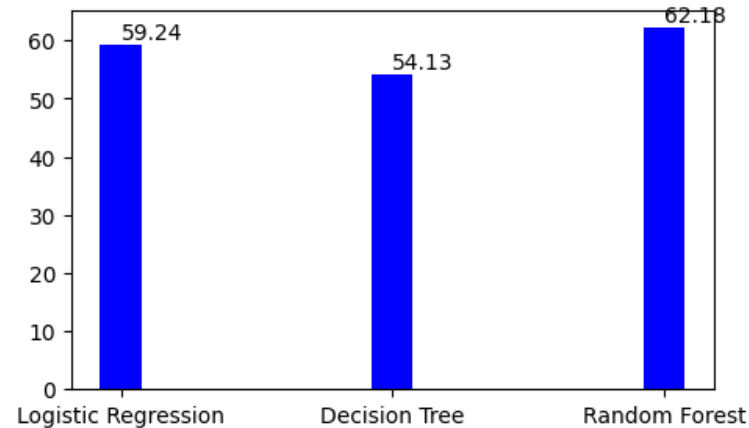
- Captures non-linear relationships but prone to overfitting with noisy data.
- Accuracy Score ~ 54%

## 3. Random Forest:

- Ensemble of trees, helped improve predictive accuracy and generalization by averaging multiple decision trees.
- Accuracy Score ~ 62%

# Model Evaluation and Final Prediction

- **Metrics:** Models evaluated based on **accuracy score** comparing predicted infection labels against true labels.



- Random Forest showed consistent improvement in accuracy ~ 62%
- **Final Prediction:** Trained Random Forest applied to test data to generate infection predictions formatted for competition submission.

# Results, Insights & Challenges

- **Numerical Summary:**
  - Balanced dataset with 50.5% infection rate.
  - Random Forest outperformed Logistic Regression and Decision Trees by 5-9% in accuracy.
- **Insights:**
  - Systems with more antivirus installations less prone to infection.
  - Geographic variations suggest region-specific threat patterns.
- **Challenges Faced:** Handling various feature types and missing data; importance of preprocessing and feature engineering.

# Conclusion

- Developed an ML model (Random Forest) to predict malware infection risk using system telemetry data.
- Achieved ~62% accuracy – higher than Logistic Regression (~59%) and Decision Tree (~54%).
- Random Forest achieved ~60% accuracy on private test data almost similar to validation data.
- Effective preprocessing was critical for performance.
- **Future scope:** Use advanced ML (e.g., XGBoost), add behavioral data, and deploy real-time threat detection.

Thank you