Malware Detection

The goal of this project was to develop machine learning models for binary classification to detect malware. The dataset contained various features extracted from executable files, such as size, entropy, and characteristics. The task was to distinguish between legitimate and malware files based on these features.

Data Preparation and Exploration: The dataset was first loaded using pandas (a software library in Python that simplifies data manipulation and analysis by providing easy-to-use data structures and tools), and initial exploration revealed that it contained 57 columns. Features like 'Name' and 'md5' were dropped since they were not relevant to the classification task. No missing data was found in the dataset, and all columns were numeric, eliminating the need for categorical encoding.

Feature Scaling and Correlation Analysis: Z-score normalization was applied to standardize the data, ensuring all features contributed equally to the models. A correlation matrix was computed to identify highly correlated features, which could impact models like logistic regression that assume feature independence.

Dimensionality Reduction using PCA: Given the high dimensionality (54 features), Principal Component Analysis (PCA) was employed to reduce the feature space while retaining 95% of the variance. This reduced the features to about 36 principal components, which were then used for modeling.

Modeling: Several machine learning algorithms were implemented and evaluated on the preprocessed dataset:

Evaluation Metrics:

Model	Accuracy	Precision	Recall	F1 Score	Cross Val Mean Accuracy
Logistic Regression	97.87%	96.98%	95.9%	96.4%	97.3%
Random Forest	99.5%	99.15%	99.3%	99.2%	98.4%
Decision Tree	99.2%	98.6%	98.9%	98.7%	97.3%
Artificial Neural Network	98.9%	N/A	N/A	N/A	N/A
Support Vector Machine	98.8%	97.9 %	98.08 %	98.03%	98.3%
Gradient Boost	99.02 %	98.49%	98.25 %	98.37%	98.6%
Gaussian Naive Bayes	52.01%	38.34%	99.8%	55.4%	50.7%

Model | Training Time (seconds)

Logistic Regression	0.301832
Random Forest	19.356451
Decision Tree	2.038171
Artificial Neural Network	63.904171
Support Vector Machine	75.327707
Gradient Boost	37.903147
Gaussian Naive Bayes	0.108968

Conclusion: All models performed well in malware classification, with Random Forest achieving the highest accuracy. PCA reduced computational load without losing predictive power. Both SVM and ANN, though resource-intensive, delivered competitive accuracy. This project effectively applied ML to detect malware from executable file features, proving algorithm effectiveness in a real world cybersecurity application.