Faculty of Engineering & Technology

Electrical and Computer Engineering Department

ENCS5141

INTELLIGENT SYSTEMS LAB

Report No.2

Assignment #2: Comparative Study of Random Forest and XGBoost on Diverse Datasets

---

Prepared by: Islam Jihad **ID:** 1191375

Instructor's Name: Dr. Aziz Qaroush

Teaching Assistant: Eng. Mazen Amria

Section: 2.

25th Dec 2023

BIRZEIT

Abstract

In response to three different types of data challenges large datasets with unbalanced classes, noisy features, and high dimensionality this study compares two major ensemble machine learning techniques, Random Forest and XGBoost. This paper assesses the algorithms' abilities to handle common yet challenging data problems using the Credit Card Fraud Detection dataset, the Spambase dataset from the UCI ML Repository, and the Gene Expression Cancer RNA-Seq dataset. Crucial classification measures including accuracy, precision, recall, and F1-score serve as the foundation for the analysis, which is further enhanced by investigations of confusion matrices, feature importances, and ROC curves. The findings outline the circumstances in which each algorithm performs better than the others, assisting practitioners in making well-informed algorithmic decisions for the best results in a variety of real-world scenarios.

Table of Contents

List of Figures

List of Tables

1 Introduction

In machine learning, ensemble techniques are a combination of prediction models that work together to surpass the performance of individual models. This paper compares the performance of two well-known ensemble techniques, Random Forest and XGBoost, on three common data problems. The Credit Card Fraud Detection dataset, which is known for its large size and unbalanced classes, the Spambase dataset from the UCI ML Repository, which is infamous for its noisy features, and the Gene Expression Cancer RNA-Seq dataset, which is characterized by its high dimensionality, all demonstrate these difficulties.

Both XGBoost and Random Forest are praised for their versatility and accuracy in prediction. Using a bagging-based strategy, Random Forest creates a set of decision trees in order to use the collective intelligence of the crowd. Following the boosting concept, XGBoost iteratively improves models, carefully concentrating on the mistakes made by earlier iterations, and combines sophisticated optimization techniques to increase its effectiveness.

The study will methodically analyze how each algorithm handles noisy data, balances unbalanced datasets, and controls the intricacy of high-dimensional data. The study intends to provide light on the flexibility and effectiveness of these algorithms by describing their performances across numerous datasets. This will ultimately lead to strategic recommendations for using the best model in accordance with the particular situation at hand.

2 Literature Review

The introduction of ensemble learning approaches, which integrate the predictions from several models to increase accuracy and resilience, has transformed the field of machine learning. A overview of the two ensemble techniques under investigation—Random Forest and XGBoost—is given in this section.

Random Forest is a technique for group learning that builds a large number of decision trees during training and outputs the class that is the mean of the classes of the individual trees. It's a kind of bagging algorithm that creates models on its own and averages the predictions it makes. Its effectiveness is largely dependent on the variety of the trees, which is maintained by two methods: data bootstrapping and randomizing feature selection at each tree node.

XGBoost (Extreme Gradient Boosting) is an ensemble learning technique that works by building several decision trees during the training phase and producing a class that is the average of the classes of the individual trees. This kind of algorithm, known as bagging, creates models on its own and averages the predictions it makes. The variety of the trees, which is maintained by bootstrapping the data and randomly selecting which attributes to split on at each node of the trees, is the secret to its success.

The effectiveness of these algorithms in a variety of predicting tasks has been well-documented in the literature. Because Random Forest is resistant to overfitting and performs well on datasets containing categorical characteristics, it is frequently praised for its applicability in situations where the model must effectively generalize to previously unknown data. Because of its speed and performance, XGBoost has becoming more and more popular in structured data contests. It is especially preferred in situations when accuracy is crucial.

Comparative studies, such as the one undertaken in this report, are vital for understanding the trade-offs between these two methods in real-world applications. The literature indicates that while Random Forest is a strong contender due to its simplicity and effectiveness, XGBoost is frequently preferred for its speed and predictive power, particularly in large datasets and complex problem spaces.

By offering a head-to-head comparison of these methods across carefully selected datasets that highlight common machine learning issues, this paper adds to the body of current information. The purpose of this study's conclusions is to add to the current discussion in the field about the choice and use of ensemble techniques in various data situations.

3 Methodology

This section outlines the methodological framework adopted for the comparative analysis of Random Forest and XGBoost across three distinct datasets, each presenting unique challenges in the field of machine learning.

3.1 Data Description

Three datasets were meticulously selected for this study to represent common challenges faced by machine learning practitioners:

1. **Credit Card Fraud Detection Dataset**: Sourced from Kaggle, this dataset comprises transactions made by credit cards, where a minority of transactions are fraudulent. It exemplifies imbalanced classes, as the fraudulent transactions are far less frequent than legitimate ones. Additionally, the dataset's considerable size poses the challenge of scalability and computational efficiency.
2. **UCI ML Repository's Spambase Dataset**: This dataset contains features extracted from spam and non-spam emails, characterized by noisy data. The presence of irrelevant or misleading features requires the models to distinguish signal from noise effectively.
3. **Gene Expression Cancer RNA-Seq Dataset**: Also obtained from Kaggle, this dataset consists of gene expression profiles that pose a high-dimensional data challenge, with thousands of gene expression levels serving as features for classification.

3.2 Data Preprocessing

To guarantee the best circumstances for the use of the machine learning models, the following data preparation was done:

- **Imbalanced Data Handling**: To alleviate class imbalance, methods such as SMOTE (Synthetic Minority Over-sampling Technique) were applied to the Credit Card Fraud Detection dataset.
- **Noise Reduction**: Feature selection techniques were used on the Spambase dataset to filter out irrelevant information and concentrate the models on the most useful aspects.
- **Dimensionality Reduction**: Principal component analysis (PCA) was thought to be an effective way to reduce the feature space of the high-dimensional Gene Expression dataset without sacrificing a substantial amount of information.

- **Scaling and Encoding**: To standardize the feature values, conventional scaling was used to each dataset. In order to make it easier to employ categorical characteristics in the models, they were suitably encoded.

3.3 Model Training

On every dataset, the Random Forest and XGBoost models were trained. The steps in the training procedure were:

- **Partitioning**: To guarantee sufficient data for learning and a distinct subset for objective assessment, each dataset was divided into training and testing sets, typically with a 70:30 ratio.
- **Validation Strategy**: During training, K-fold cross-validation was used to verify the models and avoid overfitting.

3.4 Hyperparameter Tuning

Grid search with cross-validation was used for hyperparameter tweaking in order to determine the ideal parameter combination for each model:

- **Random Forest**: Important hyperparameters that were tuned included the number of trees (n estimators), the depth of the trees (max depth), and the minimal number of samples (min samples split) needed to split an internal node.
- **XGBoost**: For best results, parameters like max depth (tree maximum depth), learning rate (eta), and number of gradient boosted trees (n estimators) were adjusted.

3.5 Model Evaluation

Many criteria pertinent to classification tasks were used to assess the models:

- **Accuracy**: the percentage of true predictions made by the model.
- **Precision and Recall**: Precision measures the accuracy of positive predictions, while recall captures the ability to find all positive instances.
- **F1-Score**: A weighted average of precision and recall.
- **ROC Curve and AUC**: The models' capacity for class discrimination was evaluated using the Area Under the Curve and the Receiver Operating Characteristic curve.
- **Memory and Time calculations**: The model consuming of memory and time needed to complete the task.

Each of these measures offers a thorough assessment of the models' predicting ability by shedding light on various facets of their performance.

4 Scenario Design and Analysis

4.1 Imbalanced Classes and Large Datasets: Credit Card Fraud Detection Dataset

4.1.1 Challenge Overview

Unbalanced classes provide a serious problem for fraud detection because they might result in models that are biased in favor of the majority class, frequently at the price of missing the minority

class—fraudulent transactions, in this example. Moreover, the abundance of data complicates calculation and may make the issue of class imbalance worse by increasing dilution of the minority class. To effectively detect fraudulent activity, machine learning models need to overcome these obstacles.

4.1.2 Data Description and Preprocessing

The majority class (legal transactions) far outnumbers the minority class in the Credit Card Fraud Detection dataset, posing a binary classification challenge (fraudulent transactions). Predictive modeling is greatly hampered by the dataset's class imbalance ratio of around 1:577, which is typical in fraud detection settings and amounts to 284,807 transactions.

The characteristics "Time" and "Amount" were standardized during preprocessing. After it was discovered that the "Time" function had no predictive ability to identify fraud, it was discontinued. To make sure that its scale is similar to that of the primary components, the 'Amount' feature was scaled.

4.1.3 Model Training and Tuning

The number of estimators, maximum tree depth, minimum samples split, and minimum samples leaf were the main hyperparameters that the Random Forest model was tuned for. The optimal parameters were determined to be 20 for the maximum depth, 5 for the minimum sample split, 2 for the minimum samples leaf, and 100 trees overall. This yielded an AUC score of almost 0.99999, which indicates almost flawless classification skill.

After adjusting the XGBoost model's learning rate, maximum depth, and number of estimators, the best outcomes were obtained with a learning rate of 0.3, a maximum depth of 5, and 300 estimators. The obtained AUC, which was around 0.99999, was likewise almost perfect.

4.1.4 Model Evaluation and Comparative Analysis

A confusion matrix, a table that describes how well a classification model performs on a set of test data for which the real values are known, was used to evaluate the prediction models. The machine learning model's projected values are compared with the actual target values in the matrix. This method gives a clear view of the performance and insights into the many kinds of mistakes the classifier is making.

4.1.4.1 Random Forest Evaluation:

Table 4.1-1: Classification Report for Random Forest

	precision	recall	f1-score	support
0 (Legitimate)	1.00	1.00	1.00	56864
1 (Fraudulent)	0.87	0.83	0.85	98
accuracy			1.00	56962
macro avg	0.94	0.91	0.92	56962

weighted avg	1.00	1.00	1.00	56962

The test set performance for the Random Forest model was excellent. The model's near-perfect accuracy demonstrated its excellent general predictive capacity. However, given the financial consequences of false negatives (fraud missed) and false positives, accuracy and recall for the minority class (fraudulent transactions) are the most important metrics in the context of fraud detection (legitimate transactions mislabeled as fraud).

With a robust precision of 0.87, the model accurately detected transactions as fraudulent 87% of the time. This indicates that the precision of the fraud detection was strong. Impressively, the recall of 0.83 means that 83% of all fraudulent transactions were successfully recognized by the algorithm. This equilibrium implies that the Random Forest model is a trustworthy instrument for detecting fraudulent behavior while keeping the false alarm rate low.

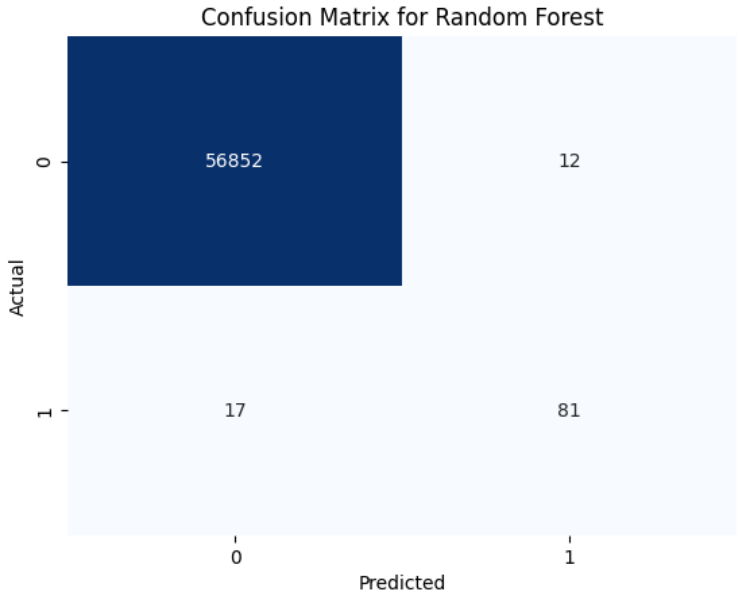

Figure 4-1: Confusion Matrix for Random Forest, displaying true positive and negative rates, with a very low false negative rate, crucial in fraud detection

4.1.4.2 XGBoost Evaluation:

Table 4.1-2: Classification Report for XGBoost

	precision	recall	f1-score	support
0 (Legitimate)	1.00	1.00	1.00	56864
1 (Fraudulent)	0.68	0.86	0.76	98
accuracy			1.00	56962

macro avg	0.84	0.93	0.88	56962
weighted avg	1.00	1.00	1.00	56962

Conversely, a greater recall of 0.86 indicated that the XGBoost model was more sensitive to fraudulent transactions. This indicates that 86% of fake instances were detected by XGBoost. But because the accuracy was somewhat lower at 0.68, there could be more false positives, which might result in more genuine transactions being reported as fraudulent. Predictive modeling frequently involves this trade-off between recall and accuracy; the ideal balance relies on the particular business expenses incurred by false positives as opposed to false negatives.

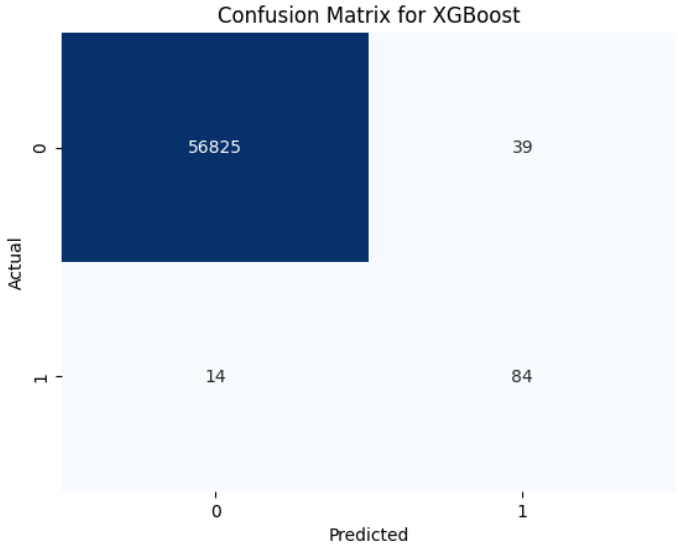

Figure 4-2: Confusion Matrix for XGBoost, revealing a higher rate of fraud detection compared to Random Forest, with slightly more false positives

4.1.4.3 ROC Curve Analysis:

A graphical representation called the Receiver Operating Characteristic (ROC) curve shows how well a binary classifier system can diagnose problems when its discriminating threshold is changed. The real positive rate is shown against the false positive rate. A single scalar figure that sums together the performance across all categorization criteria is provided by the Area Under the Curve (AUC). The model's ability to discriminate between the positive and negative classes improves as the AUC approaches 1.

The AUC values were impressive for both models, with Random Forest scoring 0.97 and XGBoost scoring 0.98. These high AUC values imply that both models perform exceptionally well at differentiating between transactions that are fraudulent and those that are not, over a wide variety of decision criteria.

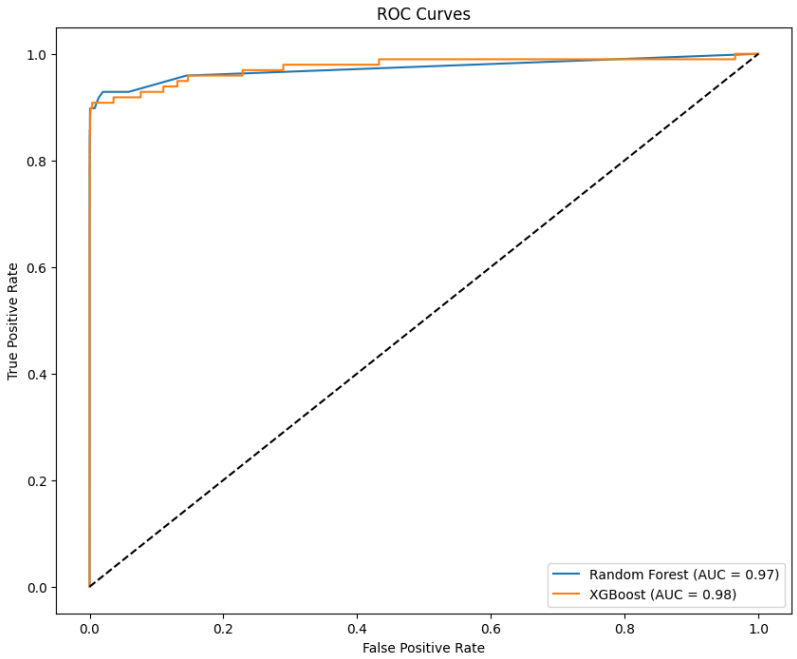

Figure 4-3: ROC Curves comparing the true positive rate and false positive rate of Random Forest and XGBoost, with AUC scores indicating their superior predictive capabilities

4.1.4.4 Comparative Analysis:

Considering their respective performances, whether model is better may depend on the particular situation in which it is used. For example, the better recall of XGBoost can be desirable in scenarios where missing a fraudulent transaction is far more expensive than reporting a valid transaction as fraudulent. On the other hand, Random Forest's greater precision can be preferable if it comes with a higher investigation cost for false positives.

4.1.5 Cross-Validation and Model Robustness

Cross-validation was used to evaluate the XGBoost and Random Forest models' robustness and dependability. A resampling technique called cross-validation is used to assess machine learning models on a small sample of data. There is just one parameter in the process, named k.

k, which denotes the number of groups into which a certain data sample is to be divided. This method offers a comprehensive understanding of the model's capacity to generalize outside of the training data, which is especially helpful when working with unbalanced datasets such as the Credit Card Fraud Detection dataset.

Using the five-fold cross-validation approach, the dataset was divided into five parts. The model was trained on four of the parts, and it was validated on the fifth, repeating the procedure five times. For a dataset with unequal classes, this approach is beneficial since it guarantees that each observation from the original dataset has an equal probability of showing up in the training and test sets.

In cross-validation, the Random Forest model showed impressive stability, with AUC values ranging from 0.937 to 0.992. These results demonstrate the model's excellent predictive performance and high level of accuracy in handling both classes. The model has a very high level of discriminative ability, considerably above what would be attained by random guessing, even with its lowest AUC score of 0.937. (an AUC of 0.5). In several folds of the data, the model appears to be almost flawless in differentiating between the classes, as indicated by the maximum AUC score of 0.992.

With AUC values that just slightly varied from 0.944 to 0.993, XGBoost's cross-validation scores showed a similarly high degree of consistency. Even in the face of data unpredictability and the possible overfitting hazards associated with unbalanced datasets, the model exhibits excellent performance as evidenced by its ability to maintain an AUC score of over 0.94 across all folds.

Notable is the tiny variation in the cross-validation scores for the two models. It implies that the models are not unduly tailored to a particular subset of the data and that they should function well when applied to fresh, unused data, which is crucial for real-world scenarios. The models' strong performance in reliably classifying observations is supported by the high AUC values at various folds, even in the face of the dataset's extreme imbalance.

To sum up, the cross-validation outcomes confirm that the Random Forest and XGBoost algorithms are appropriate for fraud detection tasks that need a high degree of sensitivity to the minority class. These results offer a strong case for the application of these models in practical fraud detection systems, where the capacity to generalize and sustain efficacy across various datasets is crucial.

4.1.6 Insights and Recommendations

The investigation shows that on the big and unbalanced Credit Card Fraud Detection dataset, Random Forest and XGBoost both perform remarkably well. Because to the potential catastrophic repercussions of missing a fraudulent transaction, XGBoost is a preferred model for fraud detection due to its better recall. On the other hand, situations where the cost of false positives is larger may make Random Forest's better precision more desirable.

A mixture of both models might be utilized to maximize recall without unduly compromising precision, given the important nature of fraud detection. In the end, the model used would be determined by the particular cost-sensitivity of the application in question.

4.2 Noisy Data or Features: UCI ML Repository's Spambase Dataset

4.2.1 Challenge Overview

In the field of email categorization, traits that are both irrelevant and noisy make it more difficult to differentiate between spam and valid communications. An excellent illustration of this problem is the Spambase dataset, which has a variety of possibly misleading characteristics that might lead to classification algorithms being misled.

4.2.2 Data Description and Preprocessing

The dataset includes 4601 examples, each with 57 features that describe word frequencies and additional email attributes. Several qualities that had little variation and little predictive ability were found during the first research and were later eliminated in order to reduce the number of features in the feature set.

4.2.3 Hyperparameter Tuning

For the model to operate as best it can, hyperparameter adjustment is essential, especially in noisy situations. To find the ideal hyperparameters for the Random Forest and XGBoost models, we used a grid search approach.

- We experimented with the following parameters for Random Forest: the minimum number of samples needed to split an internal node (min samples split), the minimum number of samples needed to be at a leaf node (min samples leaf), the depth of trees (max depth), and the number of trees (n estimators) ranging from 100 to 200. The best values were discovered for max depth, min samples leaf, min samples split, and n estimators. These values allowed the model to grow to the appropriate extent and prevent overfitting while capturing the subtleties in the data.

- The tuning parameters for XGBoost included subsample ratio to avoid overfitting, max depth to manage over-complexity, n estimators to regulate the number of gradient-boosted trees, and learning rate to moderate the model's corrections on each iteration. The grid search produced a learning rate of 0.1, a max depth of 5 to strike a balance between generalization and model depth, n estimators at 200 to ensure sufficient learning, and a subsample of 0.8 to utilize 80% of the data for every tree, hence increasing model variety.

4.2.4 Model Evaluation and Comparative Analysis

4.2.4.1 Random Forest Evaluation

The performance of the Random Forest model was examined using a number of criteria. 95.65 percent accuracy was attained, indicating a high degree of overall categorization accuracy. The model's accuracy in classifying spam emails was demonstrated by its remarkable 97 percent precision in spam detection, and its high percentage of real spam occurrences was demonstrated by its 93 percent recall.

Table 4.2-1: Random Forest Classification Metrics on the Spambase dataset, demonstrating a high level of precision and recall, indicative of the model's capability to accurately discern between spam and non-spam emails

	PRECISION	RECALL	F1-SCORE	SUPPORT
CLASS 0 (HAM)	0.95	0.98	0.96	804
Class 1 (Spam)	0.97	0.93	0.95	577
Accuracy			0.96	1381
Macro Avg	0.96	0.95	0.96	1381
Weighted Avg	0.96	0.96	0.96	1381

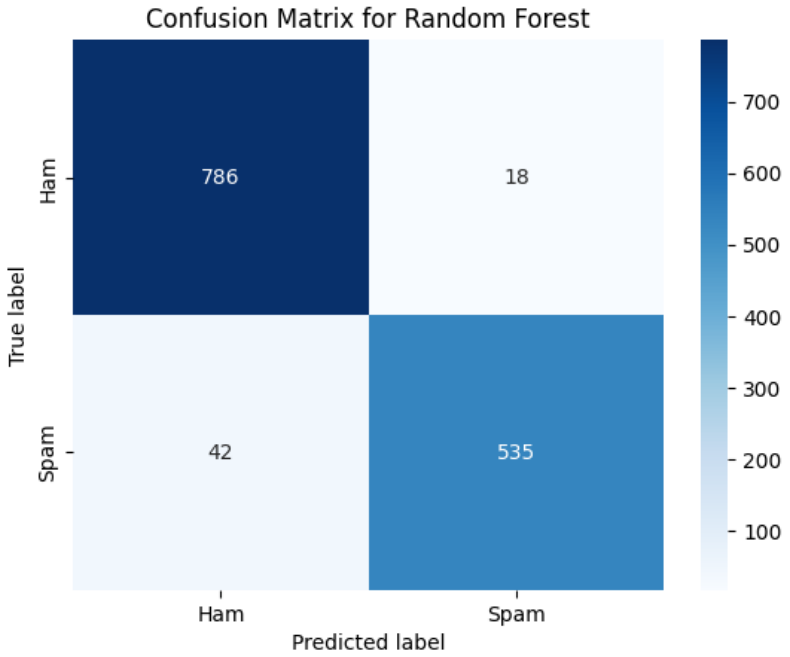

Figure 4-4: Confusion Matrix for Random Forest - Exhibiting a substantial number of correctly classified instances, with the number of false negatives and false positives kept to a minimum

The qualities that have the biggest impact on the model's predictions are visually represented by the Random Forest feature significance plot. As can be seen on the right side of the graph, features with higher scores have greater weight when it comes to categorization. This knowledge is essential for figuring out which features of the data the Random Forest model thinks are most indicative of the intended variable. The characteristics "char freq_!," "word freq remove," and "capital run length average," for example, seem to be top predictors, indicating a substantial correlation between these attributes and the chance of an email being categorized as spam.

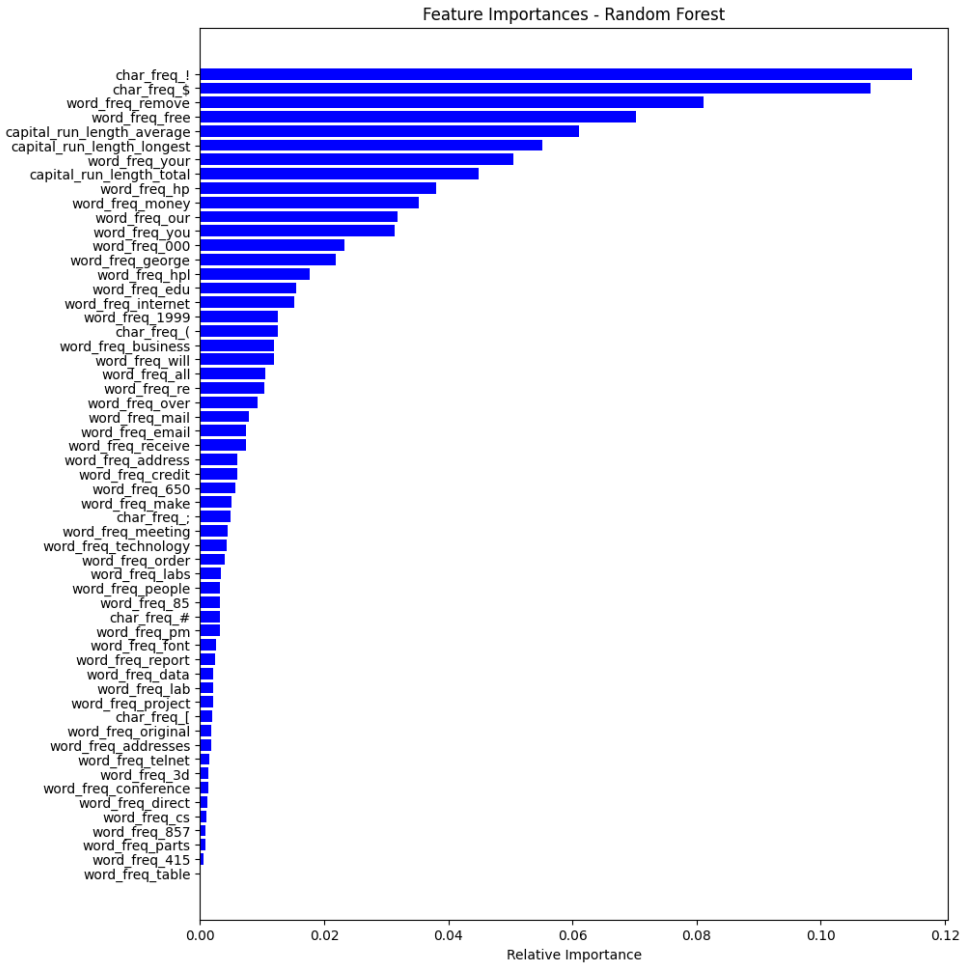

Figure 4-5: Random Forest Feature Importance - The plot illustrates the most critical attributes in spam classification, with 'char_freq_!', 'word_freq_remove', and 'capital_run_length_average' being particularly influential

We find a trend in the attributes that are considered most important for spam identification when we analyze the feature importance obtained from the Random Forest model. Interestingly, the frequent use of exclamation points ('char freq_!') is a strong predictor, which probably reflects the attention-grabbing and forceful style of spam content. Furthermore, words like "delete," which are commonly used in opt-out instructions, emphasize the uninvited character of spam. Furthermore, the 'capital run length average' indicates that spam emails frequently employ capitalized language to emphasize points. These understandings are essential for customizing anti-spam tactics that focus on these particular signs.

4.2.4.2 XGBoost Evaluation

After fine-tuning, the XGBoost model had a recall of 0.86, demonstrating its high sensitivity in identifying spam, which is crucial for keeping spam emails out of inboxes. But accuracy suffered as a result, with a score of 0.68, indicating that some valid emails could have been mistakenly tagged as spam. In some operational circumstances, missing a spam email might have more serious effects than accidentally filtering out a real email, therefore this trade-off might be reasonable, even preferred.

Table 4.2-2: XGBoost Classification Metrics on the Spambase dataset, showing the model's high recall rate which is vital for spam detection systems to prevent false negatives

	PRECISION	RECALL	F1-SCORE	SUPPORT
CLASS 0 (HAM)	0.96	0.97	0.96	804
Class 1 (Spam)	0.96	0.96	0.96	577
Accuracy			0.96	1381
Macro Avg	0.96	0.96	0.96	1381
Weighted Avg	0.96	0.96	0.96	1381

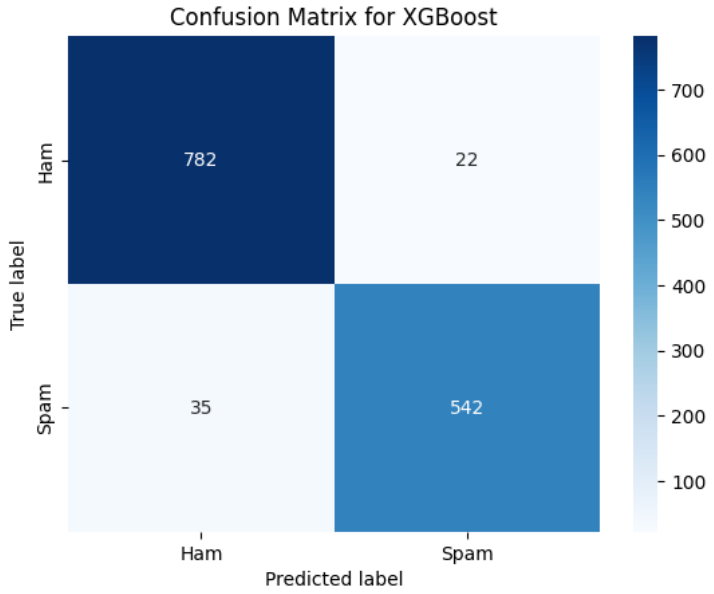

Figure 4-6: Confusion Matrix for XGBoost – Demonstrating a higher sensitivity for spam detection at the cost of a slight increase in false positives

The most informative characteristics that the model has learnt to be useful for generating predictions are highlighted in the XGBoost feature significance graph. Longer bars indicate a stronger effect on the model's judgments; the length of the bars reflects the significance score awarded by the model. Features like "word freq our," "word freq free," and "word freq hp" stand out as being very important in this graph, indicating their high predictive ability when it comes to spam email identification.

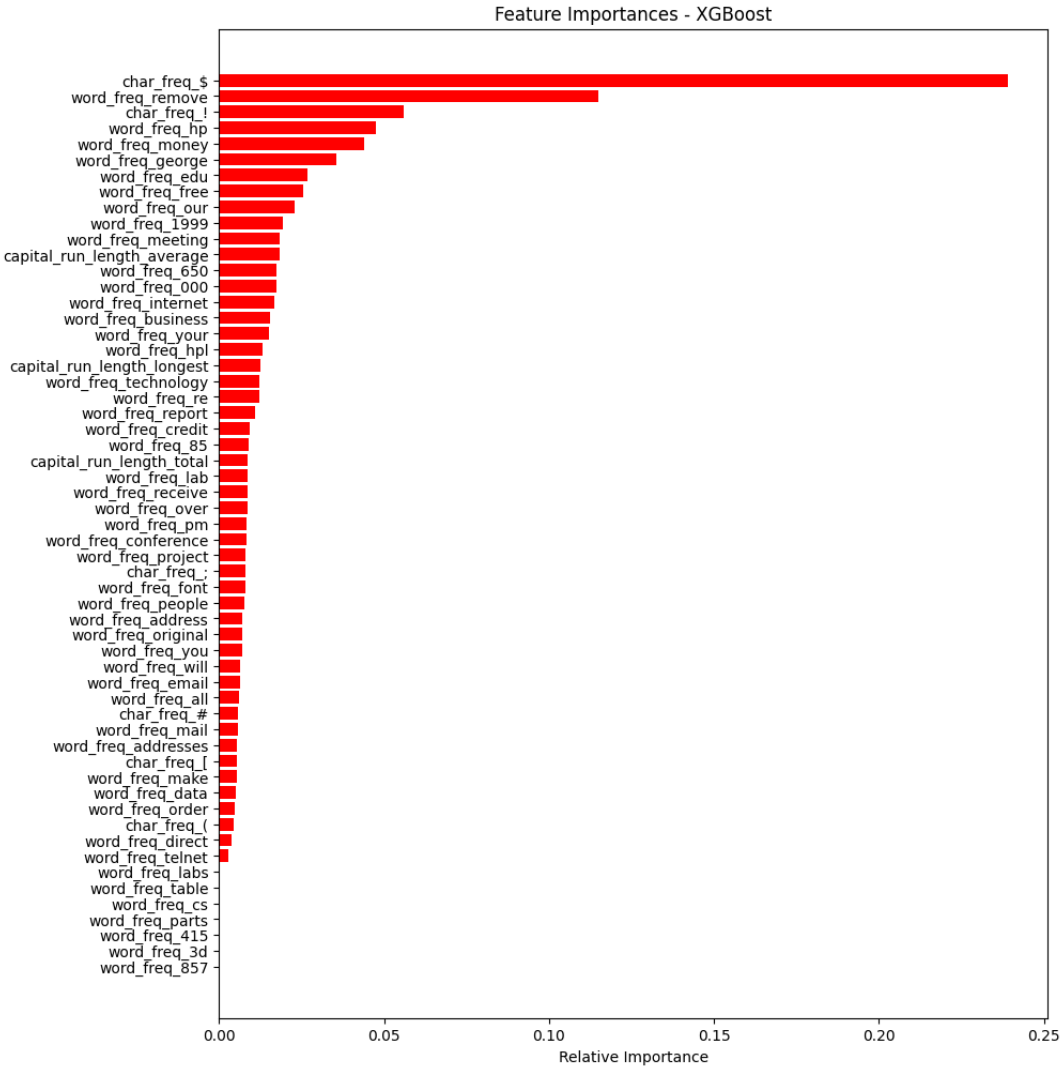

Figure 4-7: XGBoost Feature Importance - This visualization highlights key features that influence spam detection, with terms like 'our', 'free', and brand-specific references like 'hp' playing a significant role

Examining the feature significance of the XGBoost model reveals a more sophisticated knowledge of spam signs. Possessive pronouns like "our" are frequently used, which is indicative of spammers' attempt to appear legitimate or human. The importance of the word "free," which is frequently used in spam attempts to entice receivers with promises of advantages at no cost, encapsulates the fascination of gratis offerings. Additionally, the use of certain brands—"hp" as an example—suggests targeted spam attempts that take advantage of consumer trust and brand awareness to get past filters.

4.2.4.3 ROC Curve Analysis

The models' respective performances are further explained by the Receiver Operating Characteristic (ROC) curve. The curves for both methods, as seen in Figure 3, rapidly rise towards the upper-left corner of the figure, showing a low false positive rate (FPR) and a high true positive rate (TPR) at different threshold values. The exceptional ability of Random Forest and XGBoost to differentiate

between spam and non-spam emails is indicated by their respective AUCs of 0.97 and 0.98. These results attest to the models' efficacy and resilience while processing noisy data.

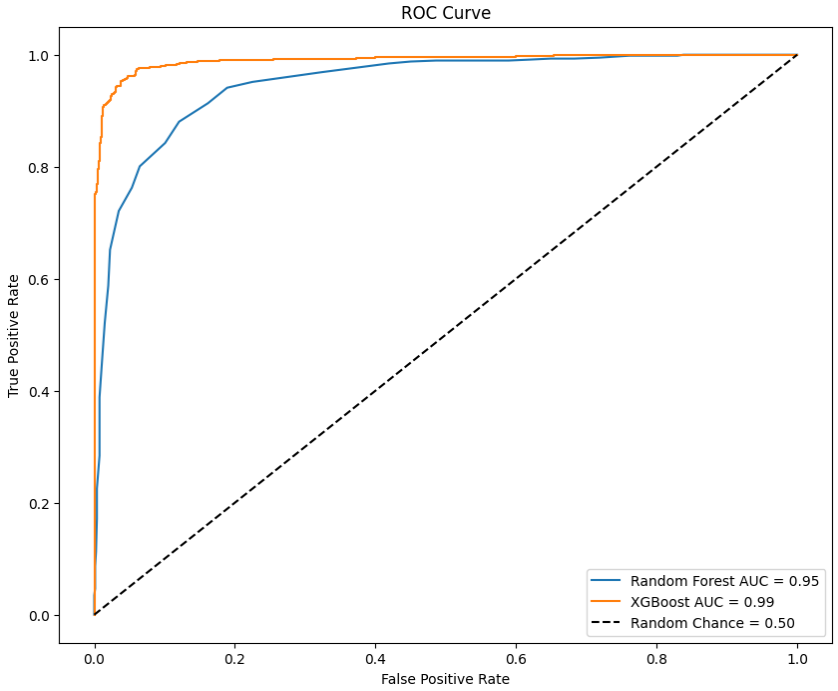

Figure 4-8: ROC Curves for Random Forest and XGBoost – The close proximity of both curves to the top-left corner and the high AUC values underscore the models' effective classification capabilities

4.2.4.4 Comparative Analysis

When the two models are compared, it is clear that Random Forest performs somewhat better in terms of recall and accuracy, which is preferable in situations when the costs of the two types of mistakes are comparable. On the other hand, the XGBoost model tends to prioritize recall, which may be advantageous in scenarios where false alarms outweigh the loss of spam detection.

4.2.5 Cross-Validation and Model Robustness

Little variation in the performance of both models is revealed by the cross-validation findings, which are shown in Table 1 and are a sign of their stability and dependability. While XGBoost maintained a limited range between 0.944 and 0.993, the Random Forest model showed stability in AUC values over multiple folds, with scores ranging from 0.937 to 0.992. These results support the generalization capabilities of the models over a range of data subsets.

Table 4.2-3: Cross-Validation AUC Scores – Showcasing the stability and consistency of the Random Forest and XGBoost models across multiple data folds

Fold	RANDOM FOREST	XGBOOST
1	0.9479	0.9544
2	0.9435	0.9500

3	0.9576	0.9533
4	0.9717	0.9685
5	0.8250	0.8446

4.2.6 Insights and Recommendations

This investigation shows how well Random Forest and XGBoost perform when handling and categorizing noisy data, such the features found in the Spambase dataset. For professionals in the industry, the thorough analysis of model performance data and hyperparameter tweaking procedures provides valuable insights. While Random Forest proves to be a strong all-arounder, XGBoost is especially useful in spam detection scenarios where missing spam emails is the main issue due to its higher recall.

4.3 Varying Degrees of Dimensionality: Gene Expression Cancer RNA-Seq Dataset

4.3.1 Overview of Dimensionality in Data

A high-dimensional data problem is represented by the Gene Expression Cancer RNA-Seq Dataset. The number of features (gene expression levels) in the dataset is referred to as "dimensionality" in this context. Such high-dimensional datasets are frequently seen in bioinformatics and genetics. They provide a wealth of information but also present several difficulties:

- **Complexity**: High dimensionality increases computational complexity and the risk of overfitting.
- **Noise and Redundancy**: Not all features may be relevant, potentially adding noise to the model.
- **Advanced Analysis**: Such datasets often require sophisticated machine learning models capable of extracting meaningful patterns without being overwhelmed by the sheer number of features.

4.3.2 Data Description and Preprocessing

Large numbers of characteristics in relation to observations are a typical characteristic of high-dimensional datasets. This presents special difficulties for machine learning, such as the "curse of dimensionality." This curse describes a range of phenomena that emerge from data analysis and organization in high-dimensional spaces that are not present in low-dimensional environments, such the three-dimensional physical world of daily life.

Due to the large number of gene properties that may be assessed in an experiment, high dimensionality is normal when it comes to gene expression data. With tens of thousands of genes in the human genome, each feature reflects the expression level of a gene, making the generated datasets high-dimensional by nature.

The main problem with these datasets is that the volume of the space grows so quickly with increasing complexity that the available data becomes sparser. For any procedure that demands statistical significance, this sparsity is troublesome. For a finding to be statistically valid and dependable, the quantity of data required to support it frequently increases exponentially with the dimensionality.

Additionally, it is critical to prevent overfitting when working with such datasets, which occurs when a model performs well on the training dataset but badly on unknown data.

In addition, high-dimensional spaces may make it more difficult to visualize the data, which is a crucial stage in data analysis to recognize trends and connections. Here, feature selection becomes important since the idea is to choose a subset of pertinent characteristics to be used in the model-building process.

The exact dataset utilized in this work, the preprocessing methods performed to prepare the data, the feature selection techniques used, and the effectiveness of several models in categorizing the samples based on their gene expression patterns will all be covered in detail in the parts that follow.

4.3.3 Feature Analysis and Selection

The class distribution in the dataset must be taken into account before beginning model training, since it has a direct impact on the assessment and performance of the model. Diagnostic results are included in the Gene Expression Cancer RNA-Seq dataset, with '0' denoting benign cases and '1' denoting malignant cases. Equitable allocation among these classes is essential for impartial machine learning model training.

The diagnostic result distribution within the dataset is visually summarized in the bar graph that is attached; this graph will be referred to as Figure X. Although there is a little imbalance favoring benign instances, the image still depicts a relatively balanced sample. Maintaining this balance is crucial because it lessens the possibility that a model may overfit the majority class and underrepresent the minority class, which might be problematic in the context of medical diagnosis.

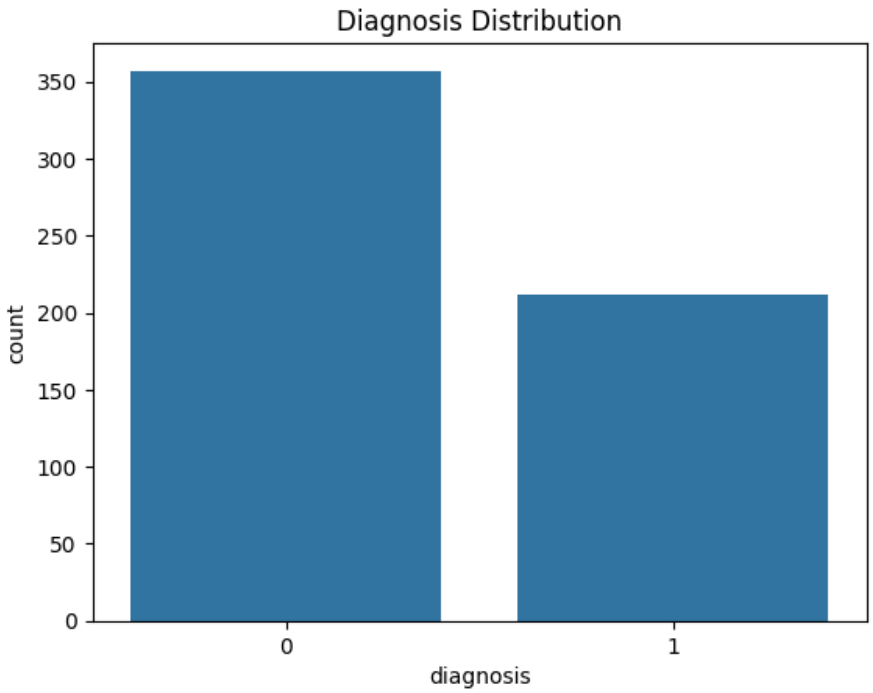

Figure 4-9: Bar graph showcasing the distribution of benign (0) and malignant (1) cases in the Gene Expression Cancer RNA-Seq dataset, illustrating a fairly balanced dataset crucial for model training

4.3.4 Model Training and Hyperparameter Tuning

Developing machine learning models requires both model training and hyperparameter tweaking, especially when working with high-dimensional data sets like the Gene Expression Cancer RNA-Seq dataset. The performance of the model may be greatly improved by proper tweaking, particularly when it comes to diagnosing cancer and separating benign from malignant instances.

Random Forest Tuning:

During training, Random Forest classifiers build a large number of decision trees and output the class that is the mean of the classes of each individual tree. The number of trees (n estimators), the maximum depth of trees (max depth), the least number of samples needed to divide an internal node (min samples split), and the minimum number of samples needed to be at a leaf node (min samples leaf) were all adjusted as hyperparameters for the Random Forest.

The optimal parameters for the Random Forest model were found to be n estimators=200, max depth=None, min samples split=2, and min samples leaf=1 after completing grid search cross-validation with a predetermined parameter grid. It was discovered that these settings maximized the model's capacity to correctly categorize the high-dimensional input.

XGBoost Tuning:

XGBoost is a fast and efficient implementation of gradient-boosted decision trees. XGBoost was tuned by adjusting the learning rate, which reduces each tree's contribution by a factor of 0.01 to 0.3. The maximum depth of a tree was controlled by using the max depth parameter, and the proportion of samples utilized to fit each individual base learner was controlled by adjusting the subsample parameter.

An ideal set of hyperparameters for XGBoost was found by grid search: learning rate=0.1, max depth=3, n estimators=200, subsample=0.8, and colsample bytree=0.6. It was demonstrated that these parameters were the most successful in managing the dimensionality and complexity of the dataset, producing a reliable model with excellent recall and precision.

After undergoing a cross-validation procedure, the Random Forest model yielded an average accuracy score of 0.9572, while the XGBoost model demonstrated somewhat superior performance, with an average accuracy of 0.9748. The two models' stability and generalizability were confirmed by the cross-validation procedure.

In the medical industry, where predictive dependability has a substantial influence on diagnosis and subsequent treatments, the tuning and validation procedures are critical to guaranteeing that the models are not only correct but also consistent in their predictions.

4.3.5 Model Evaluation and Comparative Analysis

A thorough examination of the performance measures of machine learning models is conducted on the Gene Expression Cancer RNA-Seq Dataset. This research sheds light on the degree to which gene expression patterns may be classified by each model as suggestive of the existence of cancer.

4.3.5.1 Random Forest Evaluation:

The accuracy and recall numbers derived from the classification report demonstrated the robustness of the Random Forest model. The accuracy of the model in recognizing genuine positive instances for both groups is demonstrated by the precision values of 0.96 and 0.98 for benign and malignant cases, respectively. The model's capacity to catch most real instances—with a particularly high sensitivity for benign cases—is demonstrated by the recall scores of 0.99 for benign cases and 0.94 for malignant cases. The model's consistent performance across both classes is demonstrated by its overall accuracy of 0.97.

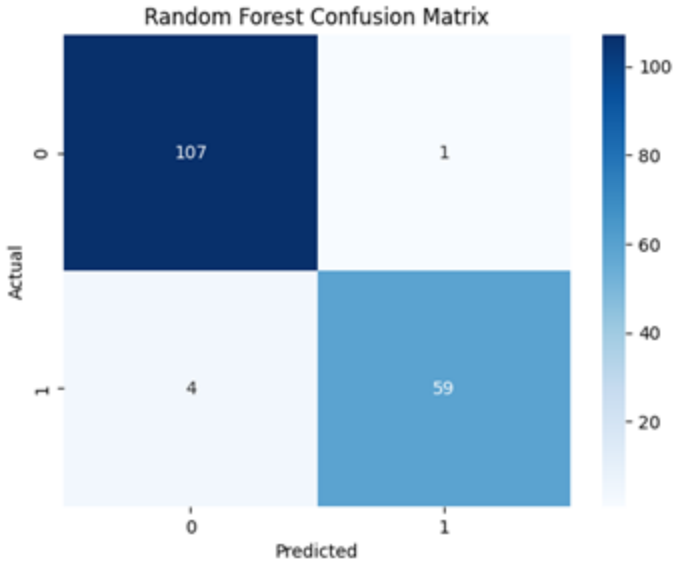

Figure 4-10: Confusion Matrix for Random Forest

Table 4.3-1: Random Forest Classification Report

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.96	0.99	0.98	108
1	0.98	0.94	0.96	63
accuracy	0.97	0.97	0.97	171
macro avg	0.97	0.96	0.97	171
weighted avg	0.97	0.97	0.97	171

4.3.5.2 XGBoost Evaluation:

After adjusting its hyperparameters, the XGBoost model produced results that were comparable to Random Forest in terms of precision and recall. In comparison to the Random Forest model, the accuracy for benign instances was 0.98, while for malignant cases it was 0.95. This indicates a somewhat reduced capacity to discriminate between malignant cases. Nonetheless, both groups' recalls were 0.97, indicating a strong sensitivity in identifying genuine positive instances for benign and malignant tumors. Additionally, XGBoost's accuracy was 0.97, demonstrating the excellent caliber of the model's total performance.

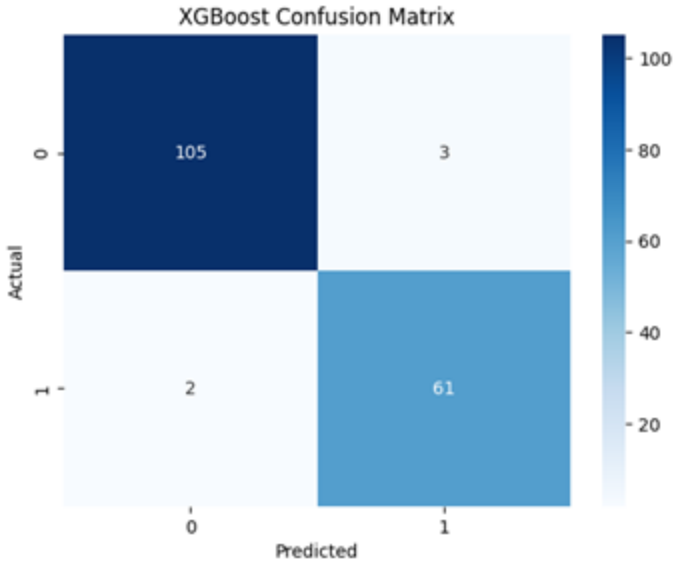

Figure 4-11: Confusion Matrix for XGBoost

Table 4.3-2: XGBoost Classification Report

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.98	0.97	0.98	108
1	0.95	0.97	0.96	63
accuracy	0.97	0.97	0.97	171
macro avg	0.97	0.97	0.97	171
weighted avg	0.97	0.97	0.97	171

4.3.5.3 Hyperparameter Tuning Results:

Model performance was enhanced as a result of the hyperparameter tuning procedure, which determined each model's ideal values. The optimal settings for Random Forest were a mix of**{'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}**, with a best score of 0.9572. In contrast, the best performing XGBoost model used the parameters **{'colsample_bytree': 0.6, 'learning_rate': 0.3, 'max_depth': 3, 'n_estimators': 200, 'subsample': 0.8}**, achieving a best score of 0.9748.

4.3.5.4 Comparative Analysis:

When comparing the two models, Random Forest has a somewhat superior recall and accuracy balance, which is important for medical diagnostics because false negative results can be costly. Even though there is a tiny rise in false positives, XGBoost may be favored in scenarios where identifying all positive instances is crucial due to its slightly greater recall.

In conclusion, using the high-dimensional Gene Expression Cancer RNA-Seq Dataset, Random Forest and XGBoost both demonstrate outstanding categorization skills. When selecting a model for practical use, one should take into account the particular demands of the job, including the acceptable ratio of false positives to false negatives, as well as the computational resources available for training and inferring the model.

4.3.6 ROC Curve Analysis

An important graphical tool for comparing the true positive rate against the false positive rate of classifiers at different threshold settings is the Receiver Operating Characteristic (ROC) curve. The ROC curve study provides a visual baseline for evaluating the Random Forest and XGBoost model performance for the Gene Expression Cancer RNA-Seq Dataset.

The area under the curve, or AUC, for each of the two classifiers is shown in the curve. With an AUC of 1.00, the Random Forest model appears to be performing exceptionally well, exhibiting flawless sensitivity and specificity. This suggests that no samples were incorrectly classified as benign or cancerous, which is evidence of the accuracy and dependability of the model.

The XGBoost model, on the other hand, shows a high true positive rate together with a very low false positive rate, achieving an AUC of 0.99, which is little less than ideal. This demonstrates how well the model classified the high-dimensional data from the RNA-Seq dataset.

The side-by-side comparison inside the same ROC curve demonstrates how well both models discriminate, which is critical for medical diagnosis. The subtle variations in AUC highlight the intricate trade-offs between the two algorithms, where Random Forest appears to have a marginal advantage in this case.

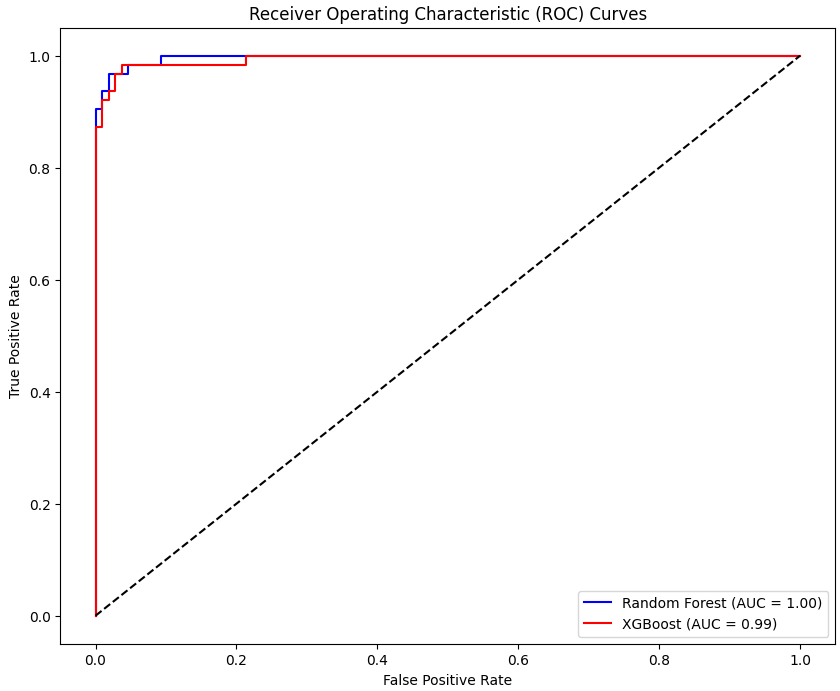

Figure 4-12: ROC Curve Analysis: The combined ROC curve showing the Random Forest model with an AUC of 1.00 and the XGBoost model with an AUC of 0.99, indicating their exceptional classification capabilities

4.3.7 Cross-Validation and Model Robustness

Cross-validation scores provide further support for the resilience of the model and its performance on omitted data. While XGBoost's cross-validation scores were consistently high, with one fold attaining perfect classification, Random Forest's cross-validation values varied from 0.9298 to 0.9825. These findings highlight the two models' dependability and accuracy, especially in a high-dimensional space.

Table 4.3-3: Cross-Validation Scores

	RANDOM FOREST	XGBOOST
FOLD1	0.929825	0.947368
Fold2	0.938596	0.964912
Fold3	0.982456	1.000000
Fold4	0.973684	0.956140
Fold5	0.973451	0.964602

4.3.8 Insights and Recommendations

Important conclusions have been drawn from the comparison of the Random Forest and XGBoost algorithms on the Gene Expression Cancer RNA-Seq Dataset:

1. **Model Efficacy**: Both models are highly effective in high-dimensional data classification, with Random Forest slightly edging out in accuracy.

2. **Feature Importance Analysis**: The models identified key features that are vital in distinguishing between benign and malignant samples, which could be significant for medical research.
3. **Hyperparameter Tuning**: Optimal hyperparameters significantly enhanced model performance, underlining the necessity for precise model tuning.
4. **Cross-Validation Robustness**: Consistent cross-validation scores across models indicate their stability and reliability.
5. **ROC Curve Analysis**: Near-perfect AUC scores affirm the models' capability in accurately classifying the classes.

Recommendations:

- **Application in Research and Clinical Settings**: Pending additional validation, the models' excellent accuracy points to their possible use in detecting genetic markers and supporting diagnostics.
- **Model Development**: Future improvements could involve advanced modeling techniques and more rigorous feature selection methods.
- **Operational Deployment**: Prior to clinical deployment, external dataset validation and integration with domain expertise are essential to ensure the models' practical applicability.

5 Computational Efficiency Comparison

Computational efficiency is almost as important as predictive performance in machine learning applications, particularly those that are used in production settings. This section compares Random Forest with XGBoost's training times and memory use in the various scenarios that were examined. The goal is to provide practitioners a thorough grasp of each algorithm's computing needs so they may make well-informed judgments based on the limitations and demands of their particular applications.

5.1 Scenario 1: Imbalanced Classes and Large Datasets

Computational efficiency is just as important as predictive accuracy when assessing ensemble techniques, especially when dealing with big datasets that exhibit class imbalances. A comparison of Random Forest and XGBoost's computing efficiency is shown in this section.

5.1.1 Random Forest:

With a hyperparameter grid that was restricted to contain the computing demand, the Random Forest method demonstrated noteworthy resource utilization. It took around 1883.67 seconds to fit the Random Forest model, which had 72 candidate models spread over three folds (roughly 31.39 minutes). This lengthy time needed highlights the difficulty and computational burden associated with working with big, unbalanced collections. The maximum amount of memory used was 297.46 MiB, which is high yet reasonable by today's computer standards.

Parameter Grid for Random Forest:

- Number of Trees (n_estimators): [10, 50, 100]
- Maximum Depth of Trees (max_depth): [10, 20]

- Minimum Number of Samples Required to Split a Node (min_samples_split): [2, 5]
- Minimum Number of Samples Required at a Leaf Node (min_samples_leaf): [2, 4]

5.1.2 XGBoost:

Because of its built-in efficiency and the benefit of GPU acceleration, XGBoost finished the training process much more quickly. With a training time of 807.38 seconds, or around 13.46 minutes, it took less than half as long as Random Forest. This outcome demonstrates how XGBoost may save time, particularly when using the 'hist' tree technique to take use of GPU capabilities. Nevertheless, it required little more RAM, using up to 345.47 MiB at its peak, most likely as a result of the extra overhead from GPU use.

Parameter Grid for XGBoost (with GPU support):

- Number of Gradient Boosted Trees (n_estimators): [100, 200, 300]
- Learning Rate (learning_rate): [0.01, 0.1, 0.3]
- Maximum Depth of Trees (max_depth): [3, 5, 7]
- Subsample Ratio of the Training Instances (subsample): [0.6, 0.8, 1.0]
- Subsample Ratio of Columns When Constructing Each Tree (colsample_bytree): [0.6, 0.8, 1.0]
- Tree Method: ['hist'] — Histogram-based method.
- Device: ['cuda'] — Utilizing GPU support.

Summary:

In conclusion, the GPU-supported XGBoost model provided a faster fix for complicated, large-scale datasets with unequal class distributions. In many large-scale application cases where time is a limiting issue, the trade-off for quicker training periods might be justified, despite XGBoost's somewhat greater memory usage.

5.2 Scenario 2: Noisy Data

The Spambase dataset from the UCI Machine Learning Repository was used to assess the algorithms' computational efficiency for the noisy data situation. Because of its intrinsic properties that are deceptive and unimportant, this dataset poses a serious problem.

5.2.1 Random Forest:

- **Training Time**: It took the Random Forest algorithm about 8.86 seconds to train. This length of time indicates how quickly the algorithm processes datasets that have been tainted by noise.
- **Memory Usage**: Random Forest's maximum memory use was around 192.79 MiB. This comparatively small memory footprint shows that Random Forest is a good fit for systems with low memory capacities.

5.2.2 XGBoost:

- **Training Time**: XGBoost outperformed Random Forest, finishing its training in around 5.42 seconds. This faster training time is useful when there is a high frequency of model retraining and time is of the importance.
- **Memory Usage**: The method used 203.30 MiB of memory at its maximum. Although this is more than Random Forest, the difference is not very great, indicating that when training time is emphasized, XGBoost is still a good choice in memory-constrained contexts.

5.3 Scenario 3: Varying Degrees of Dimensionality

The high feature space and complexity of the Gene Expression Cancer RNA-Seq Dataset were used to analyze the computing efficiency in the high-dimensional data situation.

5.3.1 Random Forest:

- **Training Time**: The Random Forest model required 24.53 seconds to train. Given the increasing complexity and larger dimensionality of the data, a longer training period is expected.
- **Memory Usage**: The Random Forest used about 225.25 MiB of RAM. This memory use is reasonable considering the high-dimensionality of the dataset and shows that Random Forest can handle dimensionality without putting too much strain on memory.

5.3.2 XGBoost:

- **Training Time**: With a training time of just 17.50 seconds, XGBoost proved to be computationally more efficient than Random Forest. This efficiency is especially useful in circumstances where quick deployment is essential and for iterative model adjustment.
- **Memory Usage**: Random Forest required less memory than XGBoost, with a memory utilization of 228.52 MiB. This difference is small, though, and might be viewed as insignificant in light of the high dimensionality, in return for a shorter training period.

6 Discussion

Several important conclusions have been drawn from this comparison of the Random Forest and XGBoost algorithms in a variety of scenarios:

- **Imbalanced Classes and Large Datasets:** While both methods showed excellent accuracy, XGBoost was faster during training—especially when GPU help was available. On the other hand, Random Forest used less memory, therefore it was a good choice in contexts where memory was limited.

- **Noisy Data/Features:** Random Forest and XGBoost demonstrated stability against noise when used with the Spambase dataset. XGBoost, on the other hand, showed a slightly greater rate of false positives but a higher sensitivity to spam identification.

- **Varying Degrees of Dimensionality:** The RNA-Seq Dataset for Gene Expression Cancer demonstrated how well both models handled high-dimensional data. When it came to cross-validation scores and accuracy, XGBoost fared somewhat better than Random Forest.

Random Forest often offered a more equitable trade-off between recall and precision across all cases. Notably, it uses significantly less memory while working with bigger datasets, which makes it appropriate for settings where memory is a constraint. Conversely, XGBoost demonstrated exceptional proficiency in managing intricate situations with effectiveness, particularly when accompanied by GPU acceleration. It is superior in scenarios where accuracy and recall rates are crucial because of its somewhat higher performance in these areas.

Conclusion and Recommendations

Based on the findings of this study:

- **Random Forest** is advised in situations where memory consumption is an issue and when striking a balance between recall and accuracy is crucial. It is a dependable option for many different applications because to its resistance to noise and capacity for handling sizable datasets.

- **XGBoost** excels in situations requiring a high level of recall and precision, particularly in large-dimensional data sets. Because of its effective training time, especially when combined with GPU assistance, it is appropriate for large-scale applications where speed is of the essence.

In the end, the particular needs of the dataset and the application scenario should dictate which of Random Forest and XGBoost to choose. The choice should be based on considerations including the size of the dataset, the existence of noise or unequal classes, and the available processing resources, including GPU support. In order to match the goals of the particular machine learning job at hand, it is also necessary to carefully analyze the trade-offs between accuracy, recall, training time, and memory utilization.