Technical Report: Liver Disease Classification Using Machine Learning

Jennifer Martinez

December 5, 2025

Abstract

Liver disease poses a major public health challenge, and early identification can significantly improve patient outcomes. In clinical practice, diagnosis often relies on biochemical tests, which can be difficult to interpret consistently. This project investigates whether machine learning models can help predict the presence of liver disease using routine clinical measurements.

Using the Indian Liver Patient Dataset (ILPD), I evaluated several supervised learning models—Logistic Regression, KNN, Decision Trees, Random Forests, Gradient Boosting, and a Neural Network—to understand how different types of models behave on small, noisy, and imbalanced medical data. I also implemented an autoencoder to explore learned low-dimensional representations and assess whether compressed features support classification.

Evaluation metrics centered on recall, precision, F1-score, ROC-AUC, and PR-AUC, since false negatives are more costly in medical settings. The Neural Network achieved the highest accuracy and F1-score, while Logistic Regression provided the best ROC-AUC and PR-AUC. Ensemble methods also performed strongly. The autoencoder-based model performed worse but provided useful insight into the dataset structure. Overall, the results highlight the strengths and limitations of classical and modern machine learning techniques when applied to real medical data.

1. Introduction

Liver disease affects millions of patients and can progress silently until reaching severe stages. Early detection allows for timely treatment and better long-term outcomes. Traditionally, liver disease assessment relies heavily on biochemical tests such as bilirubin levels, enzyme levels (SGOT, SGPT), albumin, and protein ratios. While these tests are informative, interpreting multiple markers simultaneously can be challenging. Some patterns are subtle, nonlinear, or only meaningful when multiple features interact.

Machine learning (ML) offers tools to analyze such interactions and capture hidden relationships. Rather than manually identifying patterns, ML algorithms learn from historical examples to make predictions on new data. In this project, I apply several ML models to classify patients as having liver disease or not, using the Indian Liver Patient Dataset (ILPD).

This project is designed to:

1. Apply ML techniques covered throughout the course to a real medical dataset.
2. Compare the performance of multiple algorithms and understand why some models work better than others.
3. Evaluate models using metrics that suit imbalanced medical data.
4. Explore dimensionality reduction techniques (PCA and autoencoders) for visualization and representation learning.

My goal is not only to train models but to **understand** their behaviors, limitations, and what their outputs mean in the context of medical decision-making.

2. Dataset & Preprocessing

2.1 Dataset Description

The ILPD dataset contains biochemical measurements from 579 patients after cleaning. Each patient record consists of:

Features

- **Age**: useful demographic indicator
- **Gender**: reported link to certain liver conditions
- **Total Bilirubin & Direct Bilirubin**: elevated levels often signal liver dysfunction
- **Alkaline Phosphatase, SGPT (ALT), SGOT (AST)**: enzymes associated with liver inflammation or damage
- **Total Proteins, Albumin**: relate to liver's ability to synthesize essential proteins
- **Albumin/Globulin Ratio**: decrease may indicate liver disease

The target label indicates the presence (1) or absence (0) of liver disease. The dataset is imbalanced: ~71% of samples belong to the positive class. This imbalance makes recall, precision, and PR-AUC especially important because accuracy alone can be misleading.

2.2 Data Cleaning

Several steps were required before modeling:

Handling Missing Values

A few samples contained missing values in the Albumin/Globulin Ratio column. Because the dataset is small and missingness was minimal, I removed rows with missing values. Imputing values here might distort relationships in a dataset this size.

Encoding Gender

Gender was encoded as:

- 1 = Male & 0 = Female

This ensures compatibility with ML algorithms that require numeric inputs.

Recoding the Target Variable

The original dataset labeled liver disease as 1 and healthy as 2. This was recoded to:

- 1 → 1 (disease) & 2 → 0 (no disease)

This aligns with binary classification conventions throughout the course.

2.3 Feature Scaling

I applied StandardScaler (z-score normalization) to all continuous features for models that depend on distances or gradients, such as:

- Logistic Regression
- KNN
- Neural Network
- Autoencoder

Trees and boosting models do **not** require scaling, but keeping both scaled and unscaled versions allowed me to train the appropriate models correctly.

2.4 Train/Validation/Test Split

To evaluate models fairly and tune hyperparameters, I used:

- **70% training**
- **15% validation**
- **15% test**

Using stratification ensured that the class imbalance remained consistent across splits. The validation set was used for tuning model parameters, and the test set was held out for final evaluation.

2.5 Exploratory Dimensionality Reduction

PCA (Principal Component Analysis)

PCA reduces high-dimensional data into a lower-dimensional projection while preserving the most variance possible. The 2D PCA plot revealed substantial overlap between the two classes. This suggests that linear separation in the original input space is difficult, helping explain why more flexible models perform better.

Autoencoder (Week 12 Material)

The autoencoder consisted of:

- Encoding: 10 → 6 → 3 **(latent)**
- Decoding: 3 → 6 → 10

The autoencoder compressed the original features to 3 values and attempted to reconstruct the original inputs. The latent space was used:

- For visualization
- As input to Logistic Regression to test classification performance using compressed features

This explores whether learned representations improve or weaken classification.

3. Modeling

This project applies a diverse set of machine learning models, ranging from simple linear methods to ensemble tree-based models and neural networks. Each model reflects a different learning mechanism and offers distinct advantages and limitations. By comparing these approaches, the goal is to understand not only which model performs best on this dataset, but also why certain model families excel or struggle in the context of small, noisy, imbalanced medical data.

Logistic Regression served as the baseline model. It assumes a linear relationship between the input features and the log-odds of the target class. Despite its simplicity, logistic regression is widely used in medical statistics due to its interpretability and stability. It provides insight into feature importance through learned coefficients and handles standardized clinical features well. I expected logistic regression to perform competitively because many biomedical datasets have monotonic or near-linear relationships among features.

K-Nearest Neighbors (KNN) represents a non-parametric, instance-based learning method that classifies new samples by examining the labels of their closest neighbors. While KNN can capture local patterns, it is very sensitive to feature scaling, noise, and class imbalance. Furthermore, with small datasets, the nearest neighbors may not adequately reflect broader population trends. Given these limitations, I anticipated that KNN would underperform on ILPD.

Decision Trees were implemented to examine nonlinear, rule-based learning. Decision trees recursively split the feature space to reduce impurity, offering interpretability and flexibility. However, decision trees are known to overfit, especially on small datasets where individual splits may capture noise rather than meaningful patterns. To control this, the maximum depth was restricted, though some instability was still expected.

To address the variance issues of decision trees, I included Random Forests, which average predictions from many trees built on bootstrap samples. By injecting randomness into both sample selection and feature selection, Random Forests tend to generalize better and provide feature importance insights. I expected Random Forests to outperform single trees due to their robustness.

Gradient Boosting was also included as the project's boosting model. Boosting involves training trees sequentially, where each tree attempts to correct the errors of the previous one. This creates a powerful ensemble capable of modeling subtle nonlinear interactions. Boosting often performs extremely well on structured tabular datasets, though it is more sensitive to hyperparameters and less interpretable.

A Neural Network (MLP) was added to model nonlinear relationships using learned representations. The network architecture included two hidden layers with ReLU activation and dropout to prevent overfitting. While neural networks can capture complex patterns, they are also prone to overfitting on small datasets. My expectation was that the neural network would perform well but might not surpass ensemble methods.

Finally, an Autoencoder + Logistic Regression model was trained to explore whether a compressed, learned representation could support classification. The autoencoder reduced the original 10-dimensional features to a 3-dimensional latent space. Since autoencoders are unsupervised and prioritize reconstruction rather than classification, I expected some performance degradation, but potentially useful insights into data structure.

Together, these models provide a comprehensive comparison of linear, distance-based, tree-based, boosting, neural, and representation-learning approaches.

4. Evaluation Methods

Evaluating a model on an imbalanced medical dataset requires more than simply reporting accuracy. In this project, nearly 71% of samples belong to the “disease” class, which means a naïve model predicting “disease” for everyone would achieve high accuracy but be clinically useless. Because false negatives (patients incorrectly labeled healthy) pose the highest risk, the evaluation strategy focuses on metrics that capture different aspects of classification performance, especially in the context of medical screening.

Accuracy provides a general sense of performance but can be misleading under imbalance. For example, a model with high accuracy may still miss many true disease cases if the dataset skews heavily toward the positive class.

To complement accuracy, I used **precision**, **recall**, and the **F1-score**, all of which were heavily emphasized in Week 8.

- **Recall** (sensitivity) measures how many disease cases the model successfully identifies.
- **Precision** measures how many predicted disease cases are actually correct.
- **F1** balances precision and recall, making it helpful when neither metric alone gives a full picture.

Beyond threshold-dependent metrics, I also evaluated models using **ROC-AUC** and **PR-AUC**, which examine performance across all possible classification thresholds.

- **ROC-AUC** indicates how well a model distinguishes between the disease and non-disease classes overall. A model with ROC-AUC near 0.5 fails to discriminate, whereas values closer to 1.0 indicate strong separation.

- **PR-AUC** is particularly informative for imbalanced datasets, because it focuses on the tradeoff between precision and recall. High PR-AUC values mean a model maintains good precision even when recall increases — crucial in medical screening, where we aim to identify as many true cases as possible without overwhelming clinicians with false alarms.

To complement these metrics, **confusion matrices** were computed to show exact counts of true positives, false positives, true negatives, and false negatives. These matrices make the model’s errors more transparent and help interpret how each classifier behaves beyond summary metrics.

Finally, I compared full **ROC curves** and **Precision–Recall curves** across all models and examined dimensionality reduction plots (PCA and Autoencoder latent space) to understand how the dataset’s structure influences the performance of different algorithms. This multi-step evaluation ensures that the results are not only statistically sound but also clinically meaningful.

5. Results

The performance of all models was evaluated on the ILPD test set using a combination of threshold-based and threshold-independent metrics. Since approximately 71% of the dataset belongs to the positive (liver disease) class, accuracy alone does not provide a complete picture of performance. Therefore, I rely heavily on recall, precision, F1-score, ROC-AUC, and PR-AUC, which offer more insight into how well each model handles class imbalance and identifies disease cases. These metrics are especially important in medical applications, where the cost of false negatives is high and models must balance sensitivity with precision. The following subsections present quantitative results, visual performance comparisons, and dimensionality-reduction insights to better understand how each model behaves.

5.1 Model Comparison

Table 1 presents the full evaluation results for all seven models. Across almost all metrics, the higher-performing models achieved strong recall and balanced precision, indicating that they correctly identified most liver disease cases while maintaining reasonable control over false positives.

One clear trend in the results is that several models—including Logistic Regression, Random Forest, Gradient Boosting, and the Neural Network—all achieve recall values above 0.93, with Logistic Regression, Random Forest, and the Neural Network reaching 0.95 or higher. This is particularly important in a medical context, where failing to detect disease (false negatives) may have serious consequences. Models with high recall are preferable in screening scenarios.

While recall was uniformly high among strong models, precision varied more noticeably. KNN and the Decision Tree both achieved decent recall but exhibited reduced precision and ROC-AUC, which signal more frequent false positives and poorer discrimination between the two classes. Logistic Regression, Random Forest, Gradient Boosting, and the Neural Network showed stronger tradeoffs, maintaining both good precision and recall.

The Neural Network achieved the highest accuracy (0.759) and F1-score (0.849), suggesting it best balanced the two error types. However, Logistic Regression achieved the highest ROC-AUC (0.803) and PR-AUC (0.918), which indicates that it separates the classes most consistently across thresholds. This outcome is not surprising: clinical biochemical measurements often contain quasi-linear relationships, and logistic regression excels in such settings.

Ensemble methods (Random Forest and Gradient Boosting) performed almost as well as logistic regression and the neural network. Their ROC-AUC values (0.775 and 0.751, respectively) and PR-AUC values (0.908 and 0.900) demonstrate strong capability in capturing nonlinear interactions without overfitting dramatically.

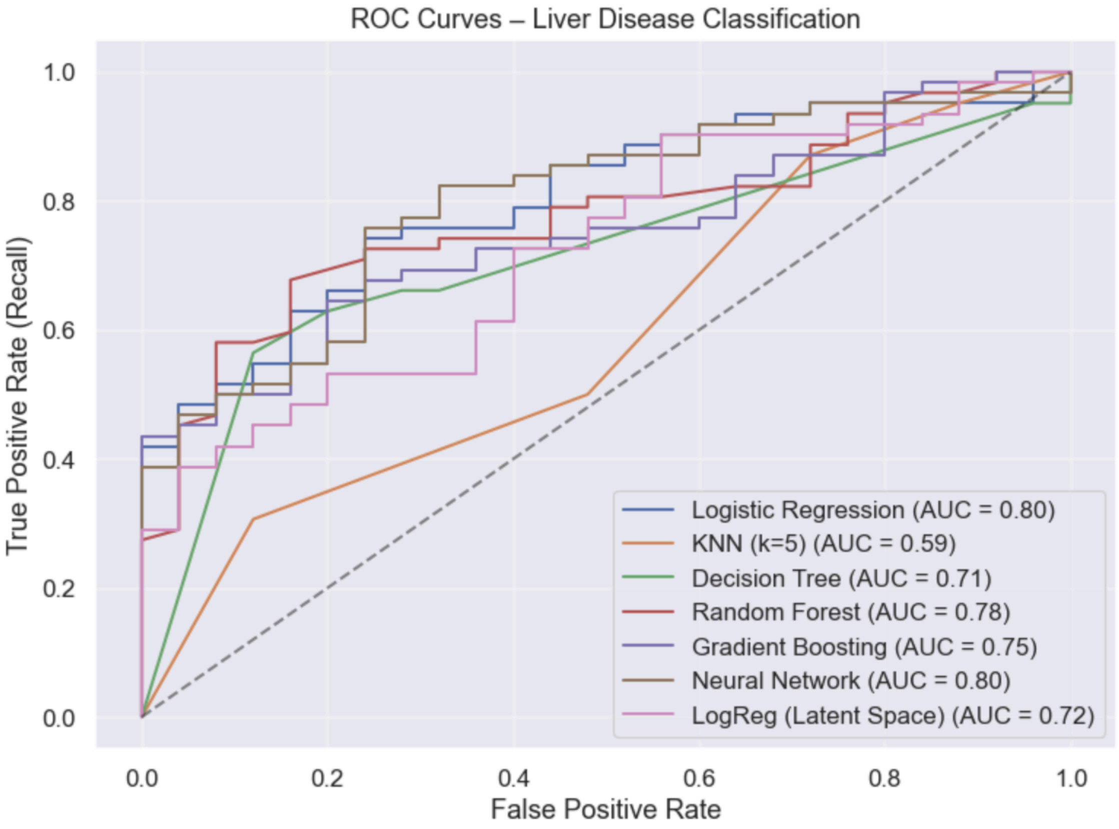
The autoencoder latent-feature model produced interesting results. It achieved extremely high recall (0.984), suggesting that the latent representation preserved almost all disease cases. However, its ROC-AUC (0.723) and PR-AUC (0.878) were lower than models trained on the original features. This supports the expectation that unsupervised compression leads to information loss, and classification performance may decline as a result. Still, the autoencoder's high recall and latent-space visualization provide meaningful insight into how the dataset behaves under nonlinear compression.

Table 1. Performance Comparison Across All Models

Model	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC-AUC**	**PR-AUC**
Logistic Regression	0.736	0.747	0.952	0.837	**0.803**	**0.918**
KNN (k=5)	0.701	0.750	0.871	0.806	0.591	0.806
Decision Tree	0.690	0.711	0.952	0.814	0.714	0.889
Random Forest	0.736	0.747	0.952	0.837	0.775	0.908
Gradient Boosting	0.724	0.744	0.935	0.829	0.751	0.900
Neural Network	**0.759**	**0.766**	**0.952**	**0.849**	0.799	0.915
LogReg (Latent Ft)	0.724	0.726	**0.984**	0.836	0.723	0.878

5.2 ROC Curve Analysis

The ROC curves show how each model balances true positives and false positives across thresholds. Logistic Regression and the Neural Network appear at the top of the ROC space, with smooth, consistently strong curves. Random Forest and Gradient Boosting follow closely behind. KNN shows a much shallower curve, emphasizing its inability to effectively discriminate between classes in this dataset. The latent-space model sits in the middle, consistent with reduced discriminative information.



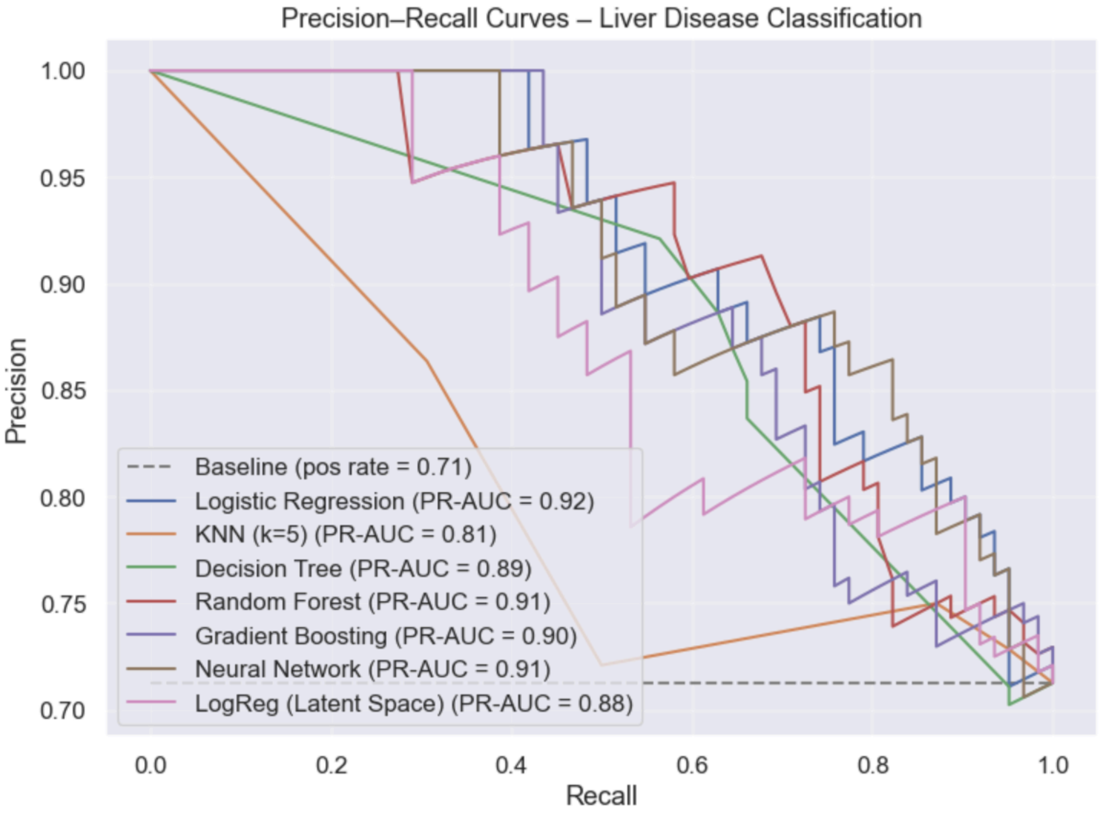
Key Observations

The ROC curves reveal distinct performance patterns across the models. Logistic Regression and the Neural Network achieved the highest ROC curves, consistent with their ROC-AUC scores near 0.80, showing strong overall discrimination between classes. Random Forest and Gradient Boosting followed closely behind, confirming that ensemble methods effectively reduce variance and capture useful nonlinear structure. The Decision Tree performed moderately well but showed signs of instability, as expected from a single-tree model. In contrast, KNN displayed significantly lower performance (ROC-AUC ≈ 0.59), suggesting it had difficulty forming reliable class boundaries in this dataset. The classifier trained on the autoencoder’s latent features produced ROC-AUC around 0.72, which is lower than the full-feature Logistic Regression model, reflecting the information loss inherent in dimensionality reduction.

5.3 Precision–Recall Curve Analysis

The PR curves reveal the practical performance under class imbalance. Logistic Regression, Random Forest, Gradient Boosting, and the Neural Network maintain precision above 0.75 even at high recall values, which is valuable for clinical screening. The Neural Network's PR curve nearly matches Logistic Regression's, demonstrating that nonlinear representational power is helpful but not overwhelmingly superior.

KNN dips earlier in the curve, meaning that as recall increases, precision falls rapidly—this is undesirable in medical settings because many healthy patients would be incorrectly flagged. The latent-feature classifier sits slightly below the top-performing models, reinforcing that compression reduces useful class-separating detail.

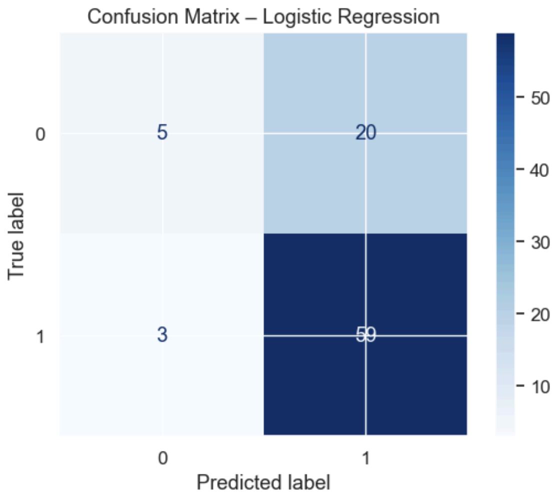


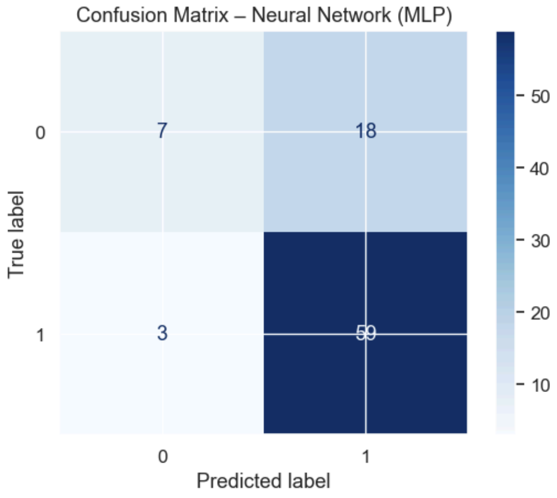
Key Observations

The PR curves reinforce several observations from the ROC curves but with more emphasis on the dataset's imbalance. Logistic Regression, the Neural Network, Random Forest, and Gradient Boosting all maintained high precision even as recall increased, producing PR-AUC values above 0.90. This indicates strong screening capability: the models are not only sensitive (high recall) but also reasonably specific (decent precision). KNN once again struggled; its precision dropped quickly at higher recall levels, showing that retrieving more true positives came at the cost of many false positives. The latent-space model achieved a respectable PR-AUC (0.878), but its reduced feature representation limited its ability to sustain precision across high recall ranges.

5.4 Confusion Matrix Analysis

Both Logistic Regression and the Neural Network produced confusion matrices with very low false negatives. This is encouraging since missed disease cases are the most harmful errors. The Neural Network generated slightly fewer false positives than Logistic Regression, resulting in marginally better specificity. These matrices help clarify how the models behave beyond numeric scores.


Confusion Matrix – Logistic Regression

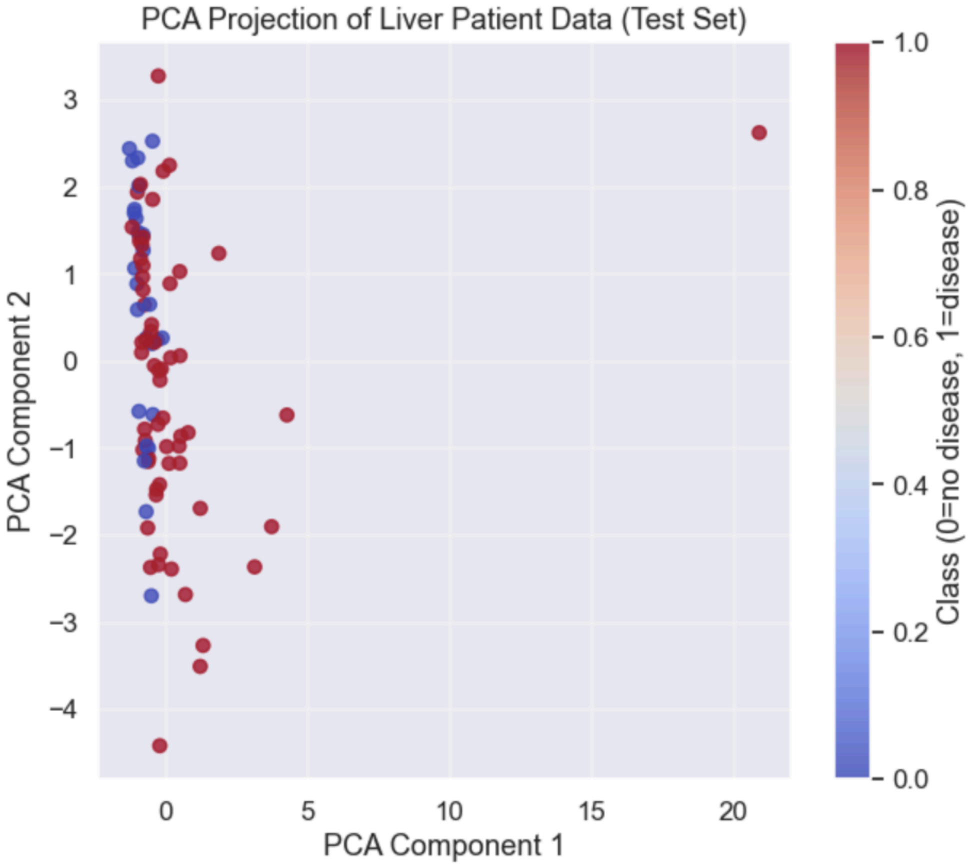

Confusion Matrix – Neural Network (MLP)

Key Observations

The confusion matrices for Logistic Regression and the Neural Network show compelling similarities and small but meaningful differences. Both models produced only 3 false negatives, which is crucial for medical screening, since missing disease cases is highly undesirable. The Neural Network slightly improved the number of true negatives, reducing false positives compared to Logistic Regression. This suggests that the Neural Network was able to improve specificity without sacrificing recall. These matrices provide reassurance that both models behave appropriately for early disease detection, identifying the vast majority of positive cases.

5.5 PCA Visualization

The PCA visualization showed substantial class overlap, meaning that there is no simple linear projection where the classes separate cleanly. This validates the challenges observed with KNN and Decision Trees and helps explain why models capable of capturing nonlinear boundaries (Neural Networks, Random Forests, Gradient Boosting) performed better. PCA's limited performance also reinforces the need to investigate nonlinear dimensionality reduction, such as autoencoders.

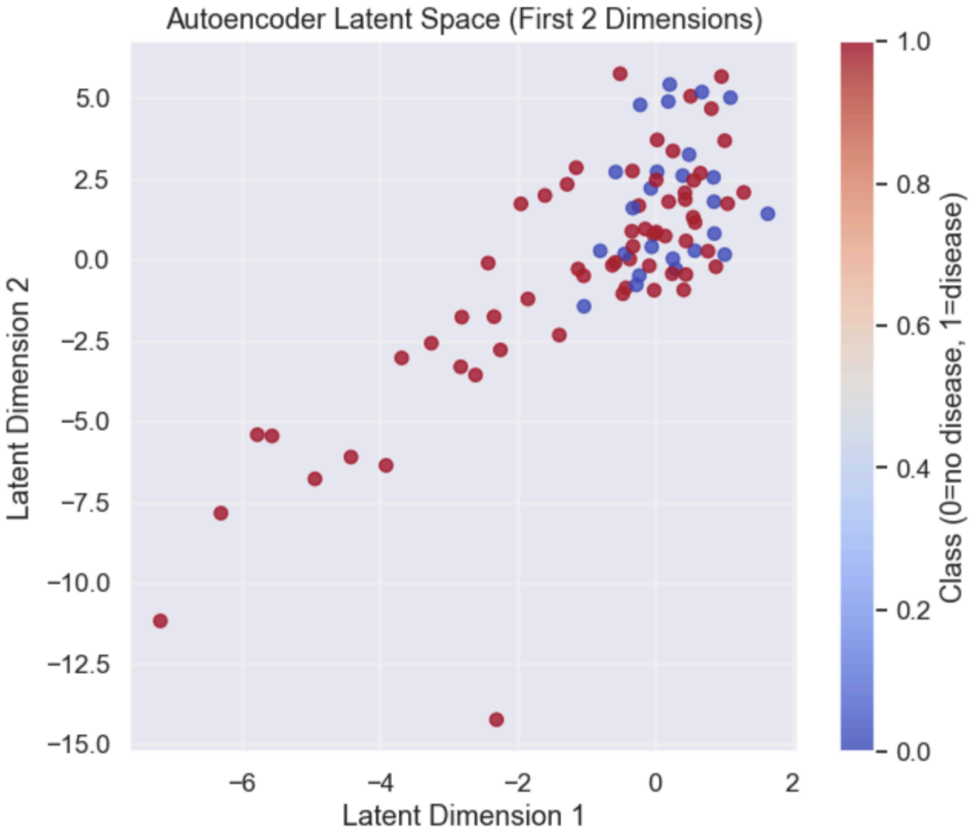


Key Observations

The PCA plot shows extensive overlap between the two classes, indicating that no simple linear projection can fully separate healthy and diseased patients. This helps explain why KNN and Logistic Regression reached a performance ceiling: the underlying feature space does not lend itself to clean linear or geometric separation. PCA does indicate some mild clustering tendencies, but these are not strong enough to form distinct decision boundaries. This motivates the need for nonlinear modeling approaches and ensemble methods, which indeed performed better in quantitative evaluations.

5.6 Autoencoder Latent Space Visualization

The autoencoder latent space displayed partial but not strong clustering between the two classes. Some structure emerged, suggesting that the autoencoder captured general patterns in the biochemical markers, but the overlap shows that unsupervised compression alone is not enough for accurate classification. These findings are aligned with the quantitative results, where the latent-feature classifier achieved good recall but lower discriminative ability.



Key Observations

The autoencoder latent space reveals subtle structure not visible in PCA, capturing more curvature and nonlinear relationships among features. However, the clusters still overlap substantially, which explains the latent model's lower ROC-AUC relative to the full-feature models. The very high recall obtained by the latent-space classifier suggests that the autoencoder retained enough information to identify positive cases reliably, but the lower precision and reduced AUC scores indicate that it compressed away class-separating detail. This reinforces the idea that autoencoders are excellent exploratory tools but not necessarily effective replacements for supervised feature learning in small datasets.

6. Discussion

This project demonstrates how different machine learning algorithms behave when applied to a small, imbalanced medical dataset. One notable outcome is that Logistic Regression, despite being the simplest model, achieved the strongest ROC-AUC and PR-AUC values. This suggests that the important relationships within the biochemical features are largely monotonic or close to linear, making logistic regression an excellent fit. In many real-world clinical datasets, linear models outperform more complex ones when the underlying patterns do not require deep nonlinear modeling.

The Neural Network performed slightly better than Logistic Regression in terms of accuracy and F1-score, likely because its nonlinear structure allowed it to capture subtle interactions between enzyme levels and protein markers. Nevertheless, the improvement over Logistic Regression was modest. This aligns with the understanding that neural networks require substantial data to significantly outperform classical methods and may overfit small datasets despite dropout.

Random Forest and Gradient Boosting also delivered strong and stable performance. Their ability to reduce variance (Random Forest) and bias (Boosting) helps them generalize well in situations where the decision tree struggles. The consistent performance of ensemble methods reinforces the importance of reducing variance when working with noisy, heterogeneous biomedical data.

The Decision Tree and KNN models highlight the challenges of applying certain algorithms to small, imbalanced datasets. The Decision Tree overfitted even with depth restrictions, and KNN's reliance on distance metrics was hampered by feature noise and overlapping class distributions. Their lower ROC-AUC values illustrate their weaknesses in this type of problem.

The autoencoder experiment provided a different perspective. Although the classifier trained on the 3-dimensional latent space did not outperform the original models, the experiment revealed how the dataset behaves under nonlinear compression. The autoencoder successfully preserved enough structure to maintain extremely high recall, but the reduced ROC-AUC and PR-AUC indicate information loss. This aligns with expectations: autoencoders are unsupervised learners that prioritize reconstruction accuracy rather than class separation.

Overall, the results suggest that simple models with good preprocessing and the right metrics can perform competitively with more complex models on medical data. This is an important takeaway for practitioners in healthcare: interpretability and stability often matter more than complexity.

7. Conclusion

This project explored a range of machine learning models for predicting liver disease using routine clinical measurements. Several models—particularly Logistic Regression, Random Forest, Gradient Boosting, and the Neural Network—performed strongly and achieved high recall, which is essential for early disease detection. Logistic Regression's competitive performance highlights how effective simple, interpretable models can be in medical applications.

Beyond numerical performance, this project revealed valuable insights into model behavior, preprocessing importance, and evaluation under class imbalance. I learned how data scaling, validation splits, and metric selection dramatically influence model outcomes. I also saw firsthand how dimensionality reduction techniques like PCA and autoencoders help illuminate underlying data structure, even when they do not improve classification accuracy.

From a broader perspective, projects like this demonstrate the potential for machine learning to support healthcare. Even small models can help screen patients, identify risk patterns, or reduce diagnostic delays when used responsibly. At the same time, the imperfections of the models remind us that machine learning should augment clinical judgment, not replace it. Improving datasets, incorporating more features, and refining algorithms may lead to tools that can meaningfully assist clinicians and improve patient outcomes.

Ultimately, this project helped connect course concepts—like bias–variance tradeoffs, ensemble learning, and neural network architectures—to a real medical dataset. It strengthened my understanding of machine learning's practical benefits and limitations and underscored the importance of thoughtful evaluation when applying ML to human health.