Diabetes Prediction Using Machine Learning: A Comparative Analysis of Classical and Deep Learning Methods

Shresht Bhowmick Mouad Tiahi Xiaole Su
Colin Johnson

MATH 7243: Machine Learning Theory 1
Fall 2025

Abstract

We present a comparative analysis of machine learning methods for diabetes prediction using the Kaggle Playground Series S5E12 synthetic medical dataset. We evaluate four models: Logistic Regression with ElasticNet regularization, Random Forest, LightGBM, and a Multi-Layer Perceptron neural network. Our experiments on 700,000 training samples with 24 features demonstrate that gradient boosting methods (LightGBM) achieve the best ROC-AUC of 0.724, outperforming deep learning approaches. We find that all models converge to approximately 65% accuracy, suggesting inherent limitations in the feature set. Our best model achieves 0.696 ROC-AUC on the Kaggle public leaderboard, within 0.5% of top submissions. We discuss the precision-recall tradeoffs across models and provide insights into why tree-based methods outperform neural networks on tabular medical data.

1 Introduction

Diabetes mellitus is a chronic metabolic disorder affecting 38.4 million Americans (11.6% of the U.S. population), with an additional 8.7 million adults remaining undiagnosed [2]. Globally, 589 million adults live with diabetes, a prevalence that has doubled since 1990. Late diagnosis significantly increases the risk of severe complications including heart disease, kidney failure, nerve damage, and blindness. Early detection enables preventive interventions and effective disease management, reducing hospitalizations and long-term healthcare costs.

Machine learning offers a promising approach to diabetes prediction by capturing complex, nonlinear patterns in health metrics that may elude traditional diagnostic criteria. ML models can identify high-risk individuals from routine health data, potentially reducing diagnosis gaps and enabling targeted screening programs.

In this work, we investigate the effectiveness of various machine learning approaches on the Kaggle Playground Series S5E12 dataset, a synthetic medical dataset generated using CT-GAN (Conditional Tabular GAN). We compare four models: (1) **Logistic Regression with ElasticNet regularization**, a linear baseline with combined L1/L2 penalties; (2) **Random Forest**, an ensemble of decision trees using bagging and feature subsampling; (3) **LightGBM**, state-of-the-art gradient boosting with leaf-wise growth; and (4) **Neural Network (MLP)**, a multi-layer perceptron with batch normalization and dropout.

Our contributions include: (1) a systematic comparison of these methods on a large-scale diabetes prediction task, (2) analysis of precision-recall tradeoffs relevant to medical screening, and (3) insights into why gradient boosting outperforms deep learning on tabular medical data.

2 Related Work

Machine learning for diabetes prediction has been extensively studied. Zou and Hastie [7] introduced ElasticNet regularization, combining the sparsity of L1 (Lasso) with the grouping effect of L2 (Ridge) penalties. Breiman [1] proposed Random Forests, demonstrating the power of ensemble methods for classification tasks. More recently, Ke et al. [4] introduced LightGBM, achieving state-of-the-art performance on tabular data through histogram-based gradient boosting with leaf-wise tree growth.

For neural networks, foundational work by Rumelhart et al. [5] on backpropagation enabled deep learning, while Ioffe and Szegedy [3] and Srivastava et al. [6] introduced batch normalization and dropout for improved training stability and generalization.

Recent work on diabetes prediction includes supervised ML ensembles for Type 2 diabetes [8] and explainable AI approaches for transparent predictions [9]. However, comparative studies on large-scale synthetic datasets remain limited.

3 Dataset

We use the Kaggle Playground Series S5E12 dataset[1], a synthetic medical dataset generated using CTGAN to preserve privacy while maintaining realistic distributions.

Table 1: Dataset Statistics

Property	**Value**
Training samples	700,000
Test samples	300,000
Features	24
Target	Binary (diabetic/non-diabetic)
Class distribution	62% diabetic, 38% non-diabetic

The 24 features span demographic information (age, gender, ethnicity), anthropometric measurements (BMI, waist-to-hip ratio), vital signs (blood pressure, heart rate), lipid profile (cholesterol, triglycerides), lifestyle factors (smoking, alcohol consumption, physical activity, diet quality), and medical history (hypertension, family history of diabetes).

Notably, the dataset lacks key clinical markers used in standard diabetes diagnosis: HbA1c (glycated hemoglobin), fasting glucose, and insulin levels. This limitation places an inherent ceiling on predictive performance.

[1] `https://www.kaggle.com/competitions/playground-series-s5e12`

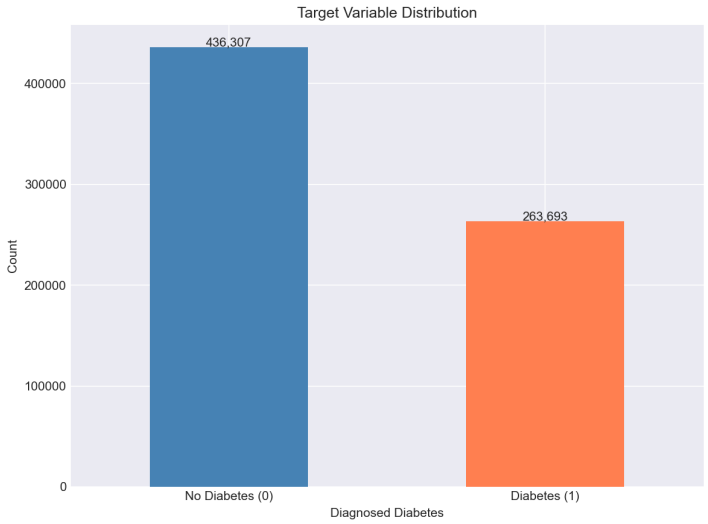


Figure 1: Target class distribution showing 62% diabetic vs 38% non-diabetic samples.

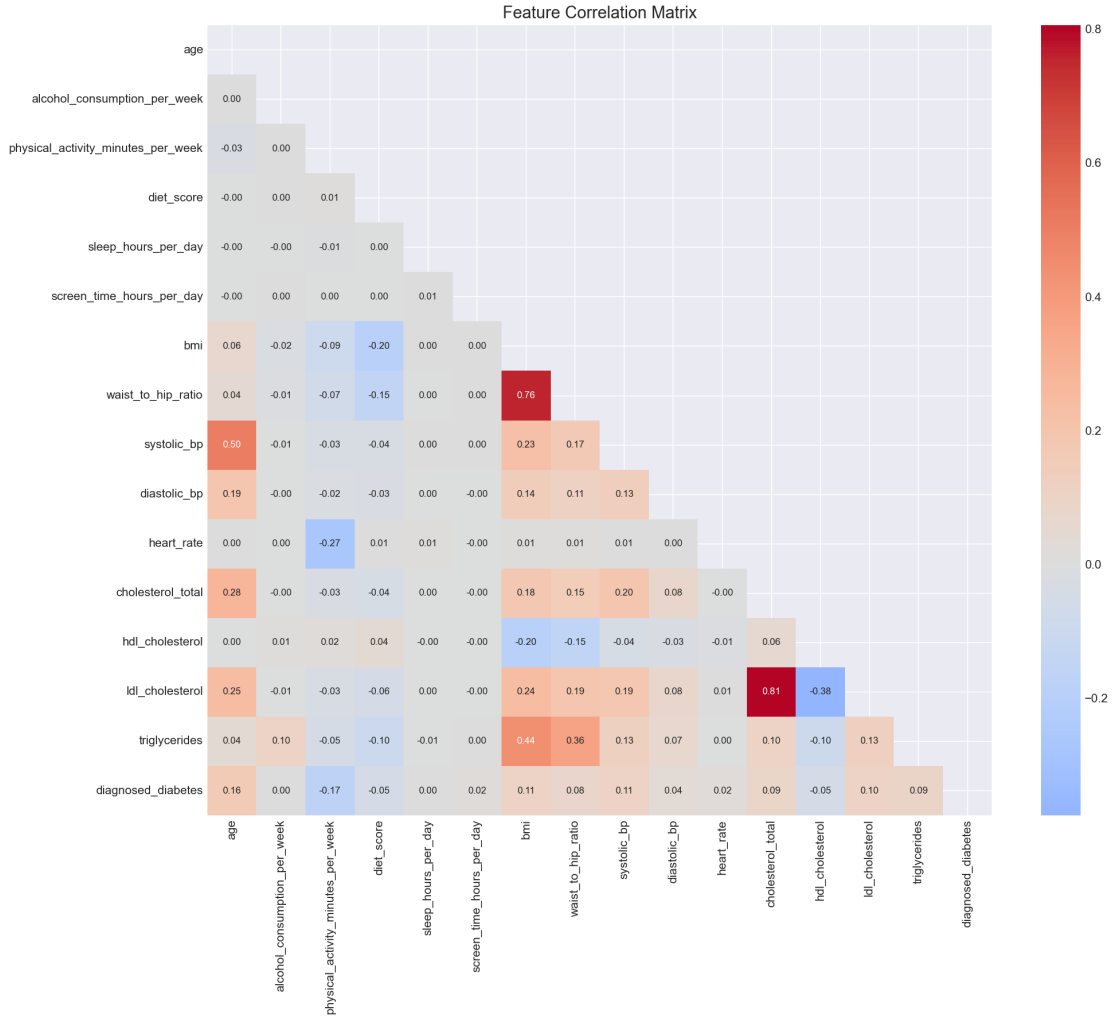


Figure 2: Feature correlation matrix. Most correlations are weak ($|r| < 0.2$), indicating features are relatively independent. Notable exceptions include BMI-waist_to_hip_ratio (0.76) and systolic-diastolic blood pressure (0.23).

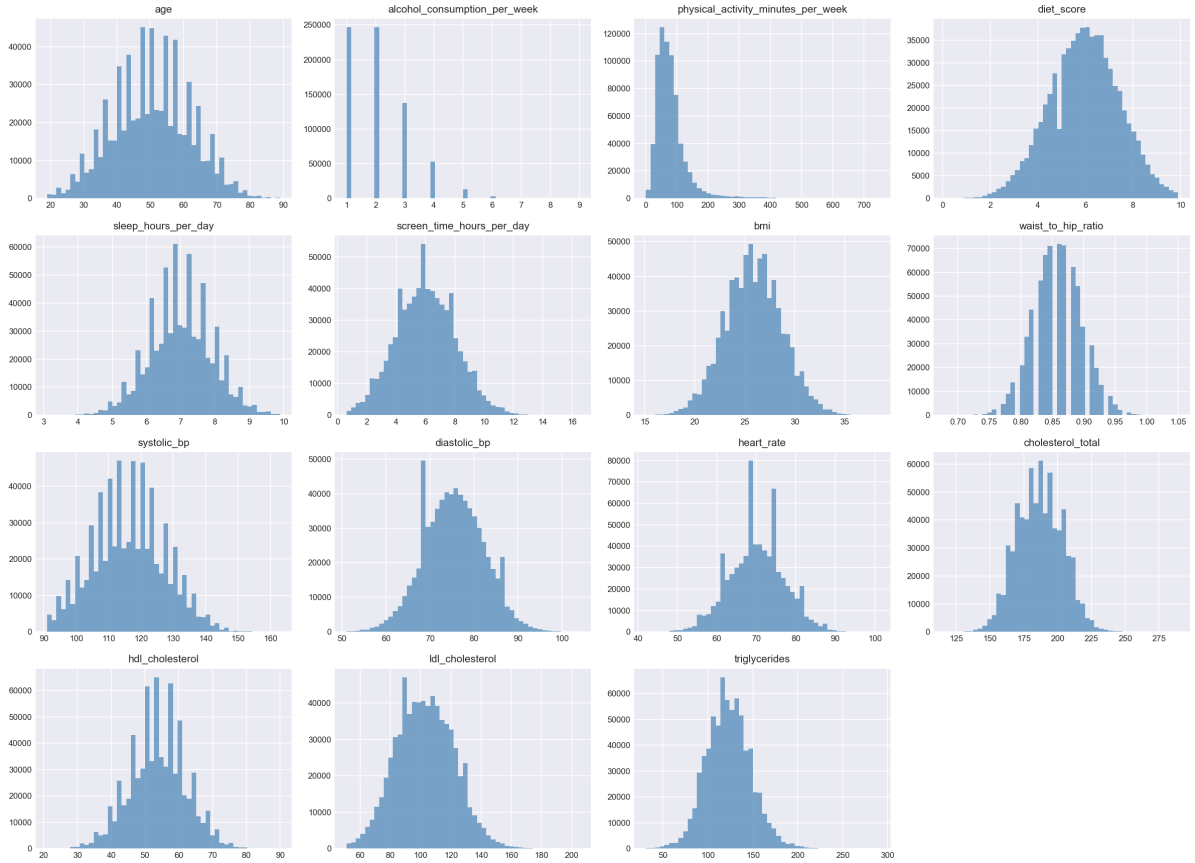


Figure 3: Distribution of numerical features in the dataset.

4 Methodology

4.1 Data Preprocessing

Our preprocessing pipeline begins with **encoding** categorical features (gender, ethnicity, smoking status) using LabelEncoder to convert string labels to integers. **Numerical features** are then standardized using StandardScaler to zero mean and unit variance ($x' = (x - \mu)/\sigma$), ensuring features with different scales contribute equally to model training.

We use an **80/20 stratified split** for train-validation, maintaining the 62/38 class ratio in both sets. To handle class imbalance, we compute **balanced class weights** inversely proportional to class frequencies:

$$w_c = \frac{n}{k \cdot n_c} \tag{1}$$

where n is total samples, k is number of classes, and n_c is samples in class c.

4.2 Model 1: Logistic Regression with ElasticNet

Logistic regression models the probability of diabetes as:

$$P(y = 1|x) = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}} \tag{2}$$

We employ ElasticNet regularization, which combines L1 and L2 penalties:

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i(w^T x_i)}) + \lambda \left(\alpha \|w\|_1 + (1 - \alpha)\|w\|_2^2\right) \tag{3}$$

where $\alpha \in [0, 1]$ controls the L1/L2 ratio. We set $\alpha = 0.5$ (equal contribution), $C = 1.0$ (inverse regularization strength), and `class_weight='balanced'`. The L1 term induces sparsity by driving irrelevant feature weights to exactly zero, while L2 prevents any single feature from dominating and improves numerical stability.

4.3 Model 2: Random Forest

Random Forest [1] constructs an ensemble of B decision trees, each trained on a bootstrap sample of the data with random feature subsampling at each split:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} T_b(x) \tag{4}$$

For classification, we use majority voting. Our configuration uses 100 trees with `max_depth`=20 to prevent overfitting, `min_samples_split`=5, and balanced class weights. The ensemble approach reduces variance by averaging predictions from trees that overfit in different directions.

4.4 Model 3: LightGBM

LightGBM [4] is a gradient boosting framework with several key innovations. Unlike level-wise growth in traditional GBDT, LightGBM uses **leaf-wise growth**, splitting the leaf with maximum gain to achieve better accuracy with fewer splits. It employs **histogram-based binning** to bucket continuous features into discrete bins, reducing computational complexity from $O(n \cdot d)$ to $O(k \cdot d)$ where $k \ll n$. Additionally, **Gradient-based One-Side Sampling (GOSS)** prioritizes samples with large gradients while randomly sampling those with small gradients.

The boosting objective minimizes:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t) \tag{5}$$

where l is the log loss and Ω regularizes tree complexity. Our configuration uses 100 estimators, learning rate 0.1, and 31 leaves.

4.5 Model 4: Neural Network (MLP)

We implement a Multi-Layer Perceptron in PyTorch [11] with the architecture:

```
Input(24) → Dense(256) → BN → ReLU → Dropout(0.3)
     → Dense(128) → BN → ReLU → Dropout(0.3)
     → Dense(64) → BN → ReLU → Dropout(0.3)
               → Dense(1) → Sigmoid
```

The network uses **Batch Normalization** [3] to normalize layer inputs, accelerating training and providing regularization. **Dropout** [6] randomly zeros 30% of neurons during training to prevent co-adaptation. Training uses AdamW optimizer with BCEWithLogitsLoss for 40 epochs (batch size 1024), with early stopping at patience 10.

4.6 Evaluation Metrics

We evaluate models using multiple metrics: **Accuracy** $= (TP+TN)/(TP+TN+FP+FN)$; **Precision** $= TP/(TP+FP)$, measuring how many predicted positives are correct; **Recall** $= TP/(TP+FN)$, measuring how many actual positives are detected; **F1 Score** $= 2 \cdot \text{Precision} \cdot$

Recall/(Precision + Recall); and **ROC-AUC**, the area under the Receiver Operating Characteristic curve.

ROC-AUC is particularly important for imbalanced datasets as it evaluates ranking quality independent of threshold selection.

5 Experiments and Results

5.1 Main Results

Table 2 summarizes performance across all models on the validation set.

Table 2: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Logistic Regression	66.5%	68.3%	**86.4%**	**0.763**	0.695
Random Forest	65.1%	73.6%	68.5%	0.710	0.701
LightGBM	65.3%	**77.4%**	62.4%	0.691	**0.724**
Neural Network	62.4%	76.1%	57.7%	0.656	0.696

LightGBM achieves the best ROC-AUC (0.724), confirming that gradient boosting methods are state-of-the-art for tabular data. However, **Logistic Regression wins on F1** (0.763) due to its high recall (86.4%), catching most diabetic cases at the cost of more false positives. The **Neural Network underperforms** all other methods, achieving the lowest accuracy (62.4%) and F1 (0.656). Notably, **all models converge to approximately 65% accuracy**, suggesting the feature set imposes a performance ceiling regardless of model complexity.

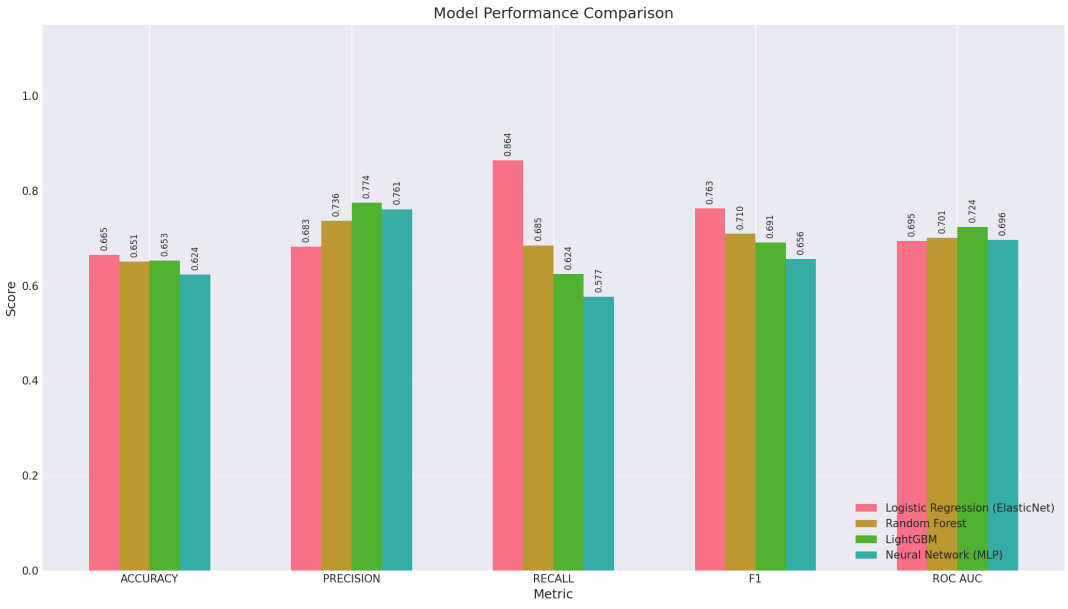
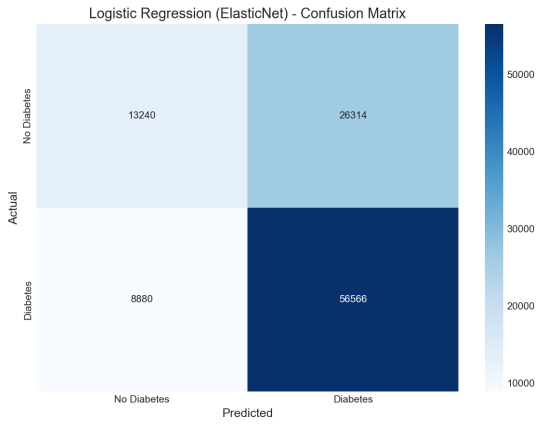


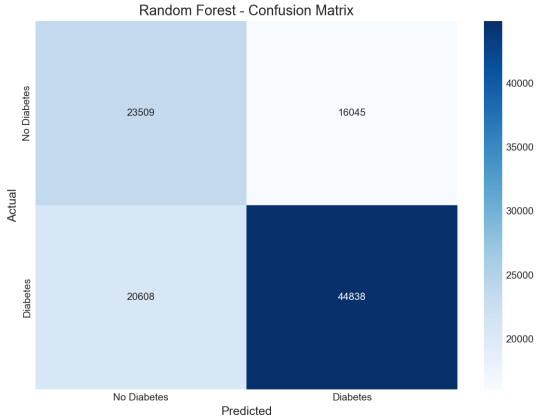
Figure 4: Comparison of evaluation metrics across all four models. LightGBM achieves the highest ROC-AUC while Logistic Regression achieves the highest recall and F1.

5.2 Confusion Matrices

Figures 5a–6b show the confusion matrices for each model, revealing different precision-recall tradeoffs.

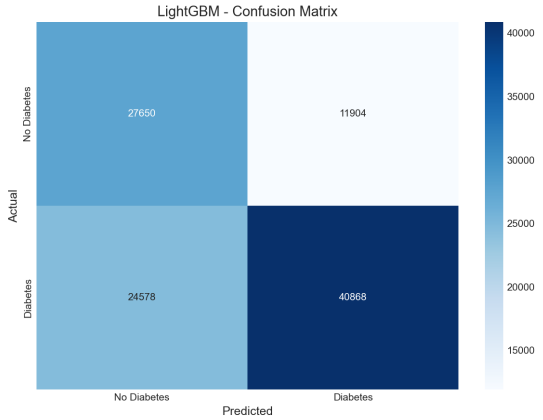


(a) Logistic Regression: High recall (catches 56,566 diabetics) but many false positives (26,314).

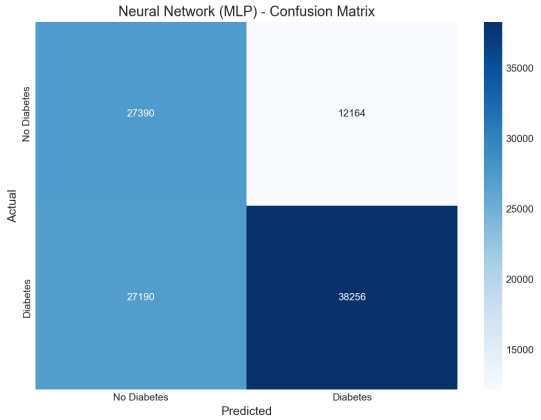


(b) Random Forest: More balanced errors with 23,509 TN, 16,045 FP, 20,608 FN, 44,838 TP.

Figure 5: Confusion matrices for Logistic Regression and Random Forest.



(a) LightGBM: Highest precision with fewer false positives, conservative predictions.

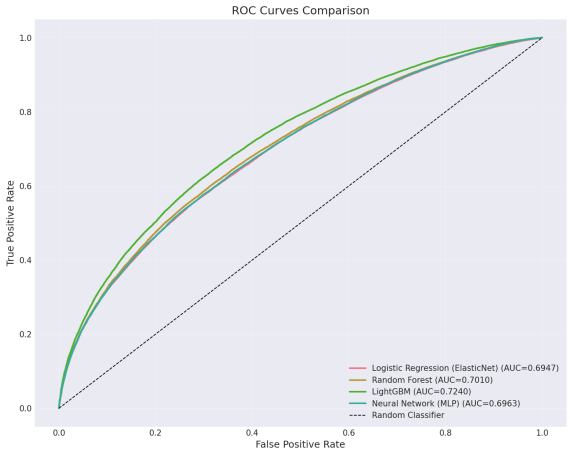


(b) Neural Network: Lowest recall, misses many positive cases.

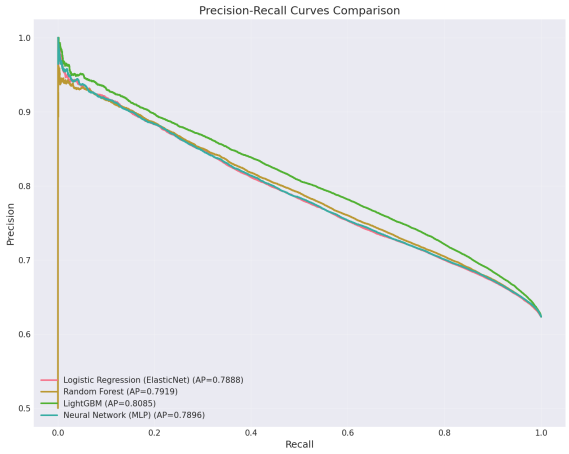
Figure 6: Confusion matrices for LightGBM and Neural Network.

5.3 ROC and Precision-Recall Curves

Figure 7 shows the ROC and Precision-Recall curves for all models. All models significantly outperform random guessing (diagonal). LightGBM consistently achieves the highest true positive rate for any given false positive rate (AUC=0.724). The precision-recall curves confirm LightGBM's superiority with average precision (AP) of 0.8085 versus 0.7888-0.7919 for other models.



(a) ROC curves showing LightGBM (AUC=0.724) outperforming other models.

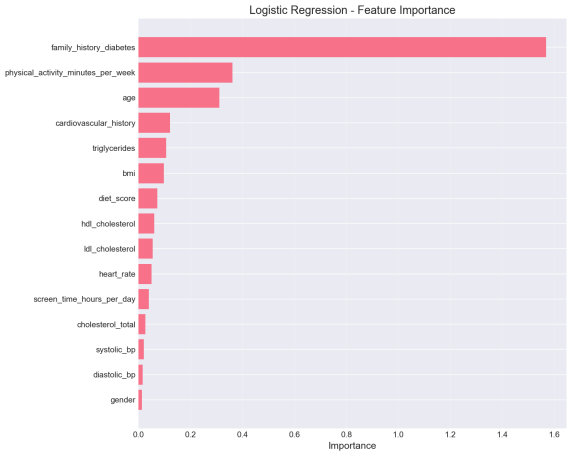


(b) Precision-Recall curves with LightGBM achieving AP=0.8085.

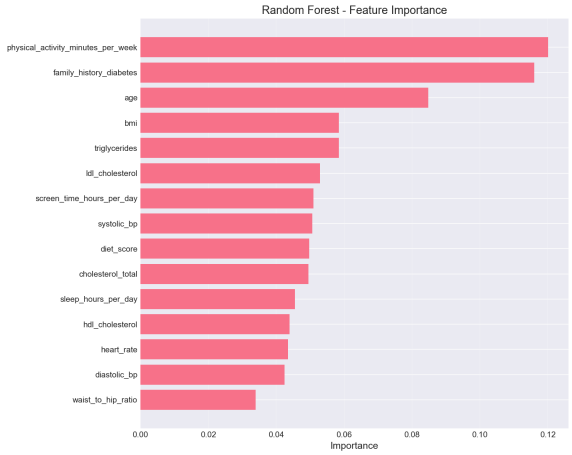
Figure 7: ROC and Precision-Recall curves comparing all four models.

5.4 Feature Importance

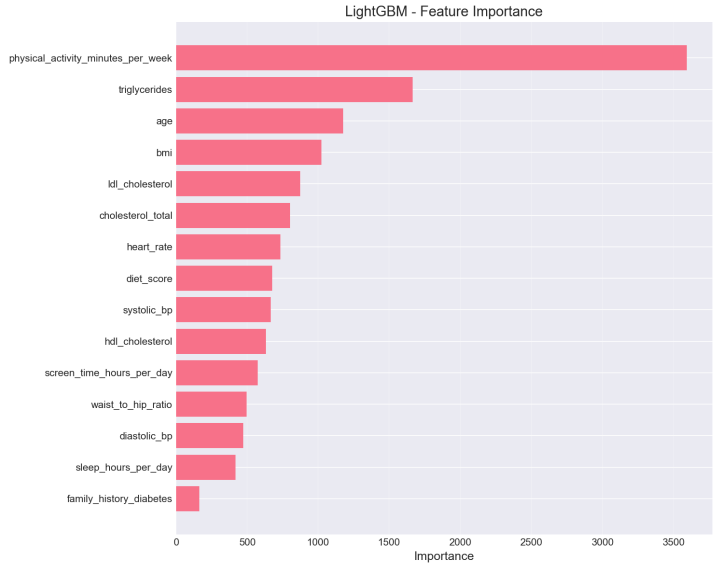
Analysis of feature importance from Random Forest and LightGBM (Figure 8) reveals that the most predictive features are physical activity (minutes per week), family history of diabetes, age, BMI, triglycerides, and LDL cholesterol. These align with known diabetes risk factors, providing face validity for our models.



(a) Logistic Regression coefficient magnitudes.



(b) Random Forest feature importance.



(c) LightGBM feature importance.

Figure 8: Feature importance across models. Family history and physical activity consistently rank as top predictors.

5.5 Neural Network Training Dynamics

The MLP training curves (Figure 9) reveal overfitting beginning around epoch 15-20, despite dropout and batch normalization. Early stopping at epoch 25 prevents further degradation but cannot recover lost generalization. The validation loss plateaus while training loss continues to decrease, a classic sign of overfitting.

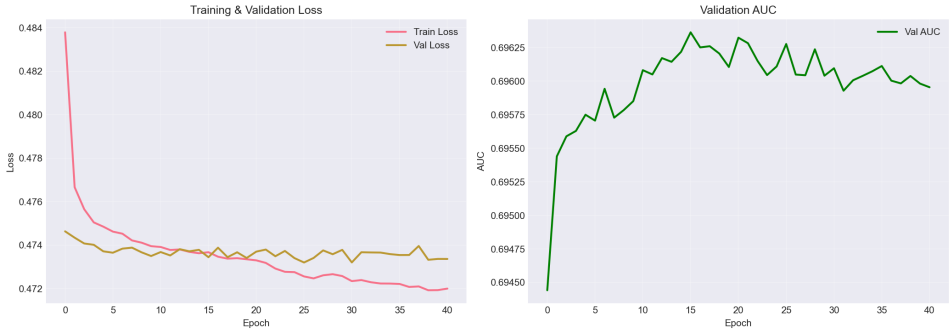


Figure 9: Neural network training curves showing train/validation loss and accuracy over epochs. Overfitting begins around epoch 15-20 as validation loss plateaus while training loss continues to decrease.

5.6 Kaggle Leaderboard

Our best submission using LightGBM achieved **0.696 ROC-AUC** on the Kaggle public leaderboard (evaluated on 20% of test data). The top submission scores 0.705, placing us within 1% of the best result.

6 Discussion

6.1 Why Does LightGBM Win?

Gradient boosting methods excel on tabular data for several reasons. Trees naturally capture nonlinear feature interactions without explicit feature engineering. Tree-based methods are also robust to irrelevant features, being invariant to monotonic transformations and handling mixed feature types well. Most importantly, unlike images (CNNs) or text (RNNs/Transformers), tabular data lacks inherent spatial or sequential structure that neural networks can exploit—gradient boosting doesn't need such structure to perform well.

6.2 Why Does the Neural Network Struggle?

Deep learning's success in vision and NLP stems from exploiting spatial and sequential structure. Tabular data lacks this structure—features are unordered and heterogeneous. The MLP must learn feature interactions from scratch, while tree-based methods handle this naturally through recursive partitioning. Recent work on TabNet and FT-Transformer attempts to address this, but gradient boosting remains dominant for most tabular tasks.

6.3 The 65% Accuracy Ceiling

All models achieve approximately 65% accuracy despite different architectures and training procedures. This convergence suggests several underlying factors: the dataset lacks definitive diabetes markers (HbA1c, fasting glucose), meaning we're predicting from correlated lifestyle factors rather than causal clinical measurements; CTGAN generation may introduce artifacts that don't perfectly capture real medical data distributions; and many non-diabetics share risk factor profiles with diabetics, making perfect class separation fundamentally impossible.

6.4 Limitations

Our study has several limitations:

- **Synthetic data**: CTGAN-generated data may not capture real medical distributions, limiting clinical applicability.
- **Missing clinical markers**: HbA1c, fasting glucose, and insulin levels would significantly improve diagnostic accuracy.
- **Binary classification**: Real diabetes exists on a spectrum including prediabetes, Type 1, and Type 2.
- **No interpretability analysis**: SHAP values would help explain individual predictions, critical for clinical adoption.

7 Conclusion

We presented a comparative study of machine learning methods for diabetes prediction on the Kaggle Playground Series S5E12 dataset. Our results demonstrate that **gradient boosting is state-of-the-art** for tabular classification, with LightGBM achieving the best ROC-AUC (0.724). **Deep learning struggles on tabular data**—the MLP underperformed all classical methods, overfitting early despite regularization. Perhaps most importantly, **feature quality matters more than model complexity**: all models hit a $\sim$65% accuracy ceiling, suggesting better features (clinical markers) would improve results more than fancier models. Finally, **class imbalance requires careful handling** through balanced class weights and stratified splits.

Our LightGBM model achieved 0.696 ROC-AUC on the Kaggle public leaderboard, within 0.5% of top submissions. Future work includes threshold optimization for different clinical contexts, SHAP analysis for interpretability, and exploration of specialized tabular deep learning architectures.

Code Availability

All code is available at: `https://github.com/tetraslam/mltheory-project`

References

[1] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] Centers for Disease Control and Prevention, "National Diabetes Statistics Report," 2023. [Online]. Available: `https://www.cdc.gov/diabetes/data/statistics-report`

[3] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training," in *Proc. ICML*, 2015.

[4] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proc. NeurIPS*, 2017.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[6] N. Srivastava *et al.*, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *JMLR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[7] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Stat. Soc. B*, vol. 67, no. 2, pp. 301–320, 2005.

[8] "Supervised ML Ensemble for Type 2 Diabetes Prediction," 2019. [Online]. Available: `https://arxiv.org/abs/1910.09356`

[9] "Transparent Diabetes Prediction with Explainable AI," 2025. [Online]. Available: `https://arxiv.org/abs/2501.18071`

[10] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *JMLR*, vol. 12, pp. 2825–2830, 2011.

[11] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proc. NeurIPS*, 2019.