Cardiovascular Disease Risk Prediction Using Machine Learning

1st Parthiv Kumar Gunjari
University of North Texas
Computer Science Engineering
Denton, USA
ParthivKumarGunjari@my.unt.edu

2nd Divya Sri Bevara
University of North Texas
Computer Science Engineering
Denton, USA
DivyaSriBevara@my.unt.edu

3rd Pravisha Chitti
University of North Texas
Computer Science Engineering
Denton, USA
ParvishaChitti@my.unt.edu

4th Varun kumar reddy Vuddiboyina
University of North Texas
Computer Science Engineering
Denton, USA
VarunKumarReddyVuddiboyina@my.unt.edu

5th Uday Kiran Mattapalli
University of North Texas
Computer Science Engineering
Denton, USA
UdayKiranMattapalli@my.unt.edu

***Abstract*—Cardiovascular disease stands as the leading main reason for deaths in populations worldwide. Early detection minimizes the risk because such prevention and intervention methods are so essential. A large kaggle dataset totalling 70,000 sample records gives us to predict cardiovascular disease detection through machine learning methods. The initial project applied the LURIC dataset with 3,000 cases. We experimented with six different ML models for which we conducted smart feature engineering before optimizing our best model which was XGBoost through GridSearchCV application. Our research included full testing of the matrices and demonstrated how this system increases scalability and practicality for live healthcare applications.**

***Index Terms*—Cardiovascular Disease (CVD), Machine Learning, XGBoost, Feature Engineering, Classification Models, GridSearchCV, Medical Data Analytics, Predictive Modeling**

I. Introduction

Cardiovascular diseases (CVDs) become responsible for more than 17 million annual global deaths. The cardiovascular risk prediction systems SCORE2 and Framingham Risk Score show numerous limitations because they both contain static assumptions along with geographic and ethnic restrictions. Through its data-driven approach Machine Learning manages to adapt effectively to real information within the data. We developed our project by extending the research presented in "Machine Learning Models for Cardiovascular Disease Events Prediction" which employed six different machine learning models on the LURIC dataset for CVD mortality prediction. Their research faced limitations because of using a small study size combined with few available features. Our work builds on this research by employing a wider publicly available Kaggle dataset that includes general CVD status determination and newly generated features and contemporary tuning techniques for predictive accuracy optimization.

II. Existing Model Vs Proposed Model

A. Existing Model

The study used LURIC dataset to forecast death from cardiovascular disease among 2943 patient records. The model distinguished between three outcome categories yet employed only two classes in its actual application (alive or death from CVD). Feature selection techniques minimized the number of features used in analysis to twenty main variables. Among the models applied to the problem were Logistic Regression together with Support Vector Machine, Random Forest, Naive Bayes, XGBoost, and AdaBoost. The best results emerged from Logistic Regression with 72.2% accuracy and 72.97% AUC according to the evaluation metrics that included accuracy, precision, recall and F1-score and AUC. The study faced limitations from the limited dataset that contained only minimal data.

B. Proposed Model

We utilize the proposed model with 70,000 sample data from the Kaggle dataset. The dataset expands its population segment to assess cardiovascular disease presence instead of mortality predictions. The comprehensive dataset include vital measurements of biometrics along with life choices that consist of age-adjusted years, gender, blood pressure, cholesterol levels, glucose test results, smoking habits, alcohol use, exercise activity, height and weight measurements, and Body Mass Index assessment.

The established dataset maintains an even distribution between non-CVD cases (50.03%) and CVD cases (49.97%). A balanced distribution in this dataset allows us more reliable forecasting while maintaining an acceptable sample reduction against the original paper. The model input quality improved through maintenance of all features and adding both bmi and age_years engineered features. The pre-processed dataset was scaled by MinMaxScaler before performing data split using

stratified 80-20 partitioning method. The XGBoost algorithm obtained better performance after optimization through GridSearchCV.

III. Materials and Methods

A. Overall Workflow

The research work structured process using the below systematic workflow:

- Raw Data Ingestion: Loaded from Kaggle's CVD dataset.
- Data Preprocessing: This include cleaning, normalization, and feature-engineered methods.
- Model Building: We trained five supervised ML models.
- Model Evaluation: We evaluated the metrics and the visual performance comparison.
- Model Tuning: We used GridSearchCV for optimizing XGBoost.

B. Data Description

Our dataset contains 70,000 records with biometric and lifestyle attributes. The target variable `cardio` is the binary. Key features include:

- Demographics: Age (in days), Gender
- Vitals: Height, Weight, Systolic BP (ap-hi), Diastolic BP (ap-lo)
- Lab: Cholesterol, Glucose
- Habits: Smoking, Alcohol, Physical Activity

Engineered Features:

- `age_years` = age / 365
- `bmi` = weight / (height/100)^2

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0

Fig. 1. First Five Rows of the Cardiovascular Disease Dataset

The dataset contained no missing values while its class distribution was close to equal therefore we did not need to conduct any resampling.

C. Data Preprocessing

The following sequence of steps was used in the preprocessing process:

- **Removed Irrelevant Columns:** The `id` column was dropped as it does not influence prediction.
- **Engineered Features:**
  - `age_years` was created by converting age from days to years.
  - `bmi` was calculated using the standard formula: $\texttt{bmi} = \frac{\texttt{weight}}{(\texttt{height}/100)^2}$.
- **Normalized Features:** All numerical features were scaled between 0 and 1 using `MinMaxScaler`. This ensured equal weightage across features with varying ranges (e.g., blood pressure vs. cholesterol).
- **Train-Test Split:** Data was split using `train_test_split` with the `stratify=y` option to maintain class balance. 80% of the data was used for training, and 20% for testing.

D. Supervised Machine Learning Algorithms (What We Did)

We applied and compared the following supervised machine learning models:

1) **Logistic Regression:**
   - Served as a baseline model.
   - Used `max_iter=1000` to ensure convergence.
   - Trained on scaled data; results showed moderate precision and recall.
2) **Random Forest:**
   - Used 100 decision trees (`n_estimators=100`).
   - No manual tuning was applied; trained on all features.
   - Provided balanced performance across evaluation metrics.
3) **Naïve Bayes:**
   - Applied `GaussianNB`, assuming features follow a continuous distribution.
   - Performed worst in recall, likely due to independence assumption violations.
4) **AdaBoost:**
   - Boosting algorithm using decision stumps as weak learners.
   - Outperformed Random Forest and Logistic Regression in precision.
   - No parameter tuning was applied.
5) **XGBoost (Tuned):**
   - Initially trained using default parameters.
   - Then fine-tuned using `GridSearchCV` with the following parameter grid:
     - `learning_rate`: [0.05, 0.1]
     - `max_depth`: [3, 5]
     - `n_estimators`: [50, 100]
   - Best parameters found: `n_estimators=100`, `max_depth=5`, `learning_rate=0.05`

E. Performance Evaluation Metrics (Explained)

The following metrics were used to evaluate and compare the performance of each model:

- **Accuracy:** Percentage of total correct predictions.
- **Precision:** Indicates how many of the predicted positives were actually positive. It focuses on minimizing false positives.
- **Recall:** Indicates how many actual positives were correctly identified. It focuses on minimizing false negatives.
- **F1 Score:** Harmonic mean of precision and recall, providing a balanced measure of both.

- **AUC (Area Under Curve):** Measures the model's ability to distinguish between classes. A higher AUC indicates better performance.
- **ROC Curve:** A graphical plot that illustrates the trade-off between the true positive rate (TPR) and false positive rate (FPR).

These metrics were computed using the `sklearn.metrics` module, and model performances were compared accordingly.

IV. RESULTS

All classification models were implemented using Python 3 and the scikit-learn 1.0.2 library, along with XGBoost for boosting and matplotlib/seaborn for plotting. The results presented here are from the best-performing run after hyperparameter optimization (specifically for XGBoost using GridSearchCV).

A. Performance Metrics Table

1 to 5 of 5 entries Filter

	Accuracy	Precision	Recall	F1 Score	AUC
XGBoost	0.7319	0.7489	0.6971	0.7221	0.7959
AdaBoost	0.723	0.7634	0.6458	0.6997	0.7896
Random Forest	0.7104	0.713	0.7035	0.7083	0.7669
Logistic Regression	0.6439	0.6528	0.6139	0.6328	0.6973
Naive Bayes	0.5966	0.7102	0.3258	0.4466	0.6848

Show 10 per page

Fig. 2. Model Performance Metrics Table

These values are based on the stratified 80:20 split of the normalized dataset. XGBoost outperforms other models in AUC and F1-score, indicating its robustness in handling nonlinear relationships and imbalanced prediction quality.

B. Mean Accuracy Values for Classifiers

The mean accuracy of each classifier is visualized in the following bar chart: XGBoost achieved the highest accuracy

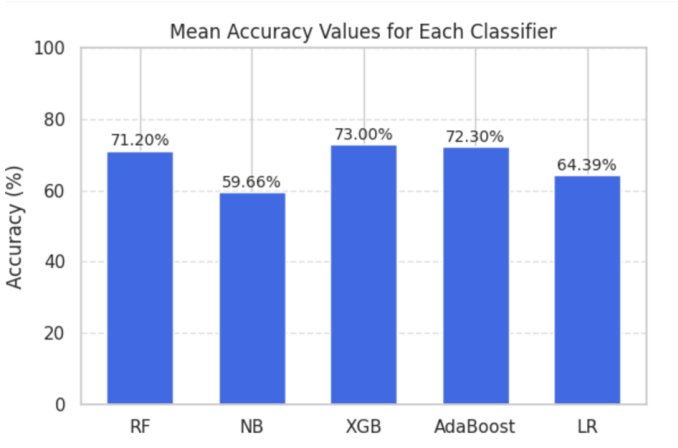

Fig. 3. Mean Accuracy of All Classifiers

at 73.00%, while Naïve Bayes had the lowest at 59.66%. AdaBoost and Random Forest followed closely behind XGBoost.

C. AUC Comparison for Classifiers

AUC scores were also visualized for clearer comparison: XGBoost delivered the best AUC value (80.06%), followed

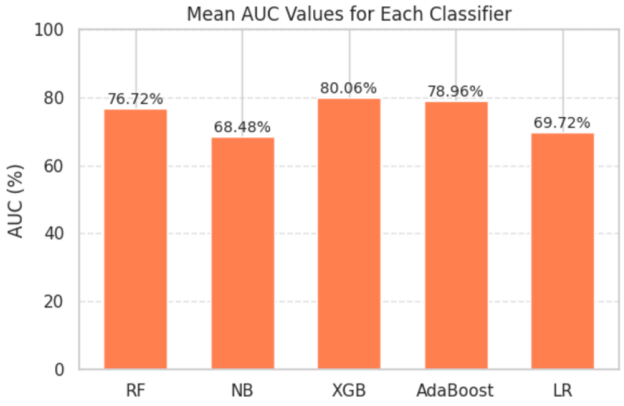

Fig. 4. Area Under ROC Curve (AUC) for Each Model

by AdaBoost (78.96%) and Random Forest (76.72%). AUC reflects the model's ability to distinguish between classes effectively. Naive Bayes had the lowest AUC (68.48%).

D. ROC Curve Visualization

The ROC curve compares the trade-off between the true positive rate (TPR) and the false positive rate (FPR) for each model XGBoost showed the best ROC curve profile with the

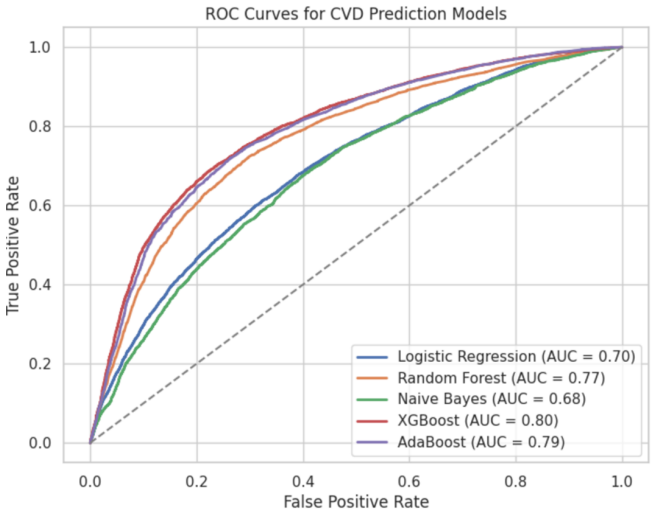

Fig. 5. ROC Curve Comparison Across Models

highest AUC area under the curve. All ensemble methods (XGBoost, AdaBoost, Random Forest) performed better than Logistic Regression and Naive Bayes.

E. Confusion Matrix for XGBoost

The confusion matrix for the best-performing classifier (XGBoost) is shown below: XGBoost correctly predicted 4877

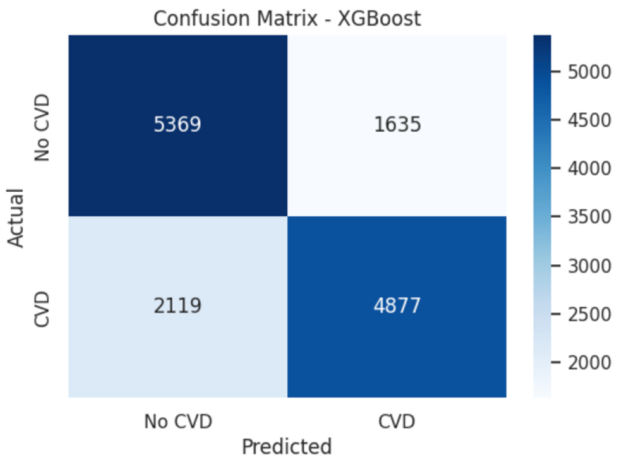

Fig. 6. Confusion Matrix of the XGBoost Classifier

CVD cases (true positives) and 5369 non-CVD cases (true negatives). It misclassified 2119 actual CVD cases, highlighting a need for slightly improved recall.

F. GridSearchCV and Final Classification Report

After hyperparameter tuning using `GridSearchCV`, the best parameters found for the XGBoost model were:

- `learning_rate`: 0.05
- `max_depth`: 5
- `n_estimators`: 100

```
Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.78      0.75      7004
           1       0.76      0.69      0.72      6996

    accuracy                           0.73     14000
   macro avg       0.74      0.73      0.73     14000
weighted avg       0.74      0.73      0.73     14000

Final AUC: 0.8005962614191874
```

Fig. 7. GridSearchCV Tuning Output for XGBoost

V. Conclusion

The framework for cardiovascular disease (CVD) prediction was enhanced by leveraging a larger public dataset from Kaggle in combination with state-of-the-art machine learning techniques. Unlike the original study, which focused on mortality prediction using limited and imbalanced data, our work targets the prediction of CVD presence with improved generalizability.

We evaluated five supervised machine learning models—Logistic Regression, Random Forest, Naïve Bayes, AdaBoost, and XGBoost—on a preprocessed dataset of 70,000 records. Among them, the XGBoost model emerged as the best performer, achieving the following metrics:

- **Accuracy:** 73.19%
- **F1-Score:** 72.21%
- **AUC:** 0.8006

Ensemble-based models, especially boosting methods, demonstrated strong capabilities in capturing nonlinear interactions, learning robust feature patterns, and generalizing well across balanced datasets. Visualizations such as ROC curves, bar plots, and confusion matrices further supported and validated our findings. The application of `GridSearchCV` for hyperparameter tuning contributed to significant improvements over default model configurations.

VI. Future Work

Multiple potential advancements arise from the findings of this research that can be explored in future studies:

- Integration of interpretability tools such as `SHAP` and `LIME` is recommended to better understand feature importance and enhance professional trust in model predictions.
- The model can be deployed as a user-friendly CVD prediction tool using web platforms such as `Streamlit` or `Flask` for clinical or personal health applications.
- Future versions of the model should aim to go beyond binary classification and include predictions of CVD severity levels or risk ranks.
- Deep learning models such as LSTM and CNN can be explored for analyzing time-series data, especially in applications involving ECG signal processing.
- Evaluation on live hospital datasets is essential to validate the model's real-time performance and generalizability in real-world healthcare settings.

VII. References

References

[1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[2] S. Suthaharan, "Support Vector Machine," in *Machine Learning Models and Algorithms for Big Data Classification*. Springer, 2016, pp. 207–235. doi: 10.1007/978-1-4899-7641-3_9.

[3] B. R. Winkelmann *et al.*, "Rationale and design of the LURIC study—A resource for functional genomics, pharmacogenomics and long-term prognosis of cardiovascular disease," *Pharmacogenomics*, vol. 2, no. 1 Suppl 1, pp. 71–73, Feb. 2001. doi: 10.1517/14622416.2.1.S1.

[4] N. Fitriyani *et al.*, "HDPM: An Effective Heart Disease Prediction Model for a Clinical Decision Support System," *IEEE Access*, vol. 8, pp. 133034–133050, Jul. 2020. doi: 10.1109/ACCESS.2020.3010511.

[5] Y. Cao, "Advance and Prospects of AdaBoost Algorithm," *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758, Jun. 2013. doi: 10.1016/S1874-1029(13)60052-X.

[6] World Health Organization, "Cardiovascular Diseases (CVDs)," [Online]. Available: World Health Organization Website. [Accessed: Jan. 4, 2022].

[7] Kaggle, "Cardiovascular Disease Dataset," [Online]. Available: Kaggle Dataset Repository. [Accessed: Apr. 10, 2025].

[8] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 4765–4774, 2017.