Journal of High School Science

Machine Learning models for cardiovascular disease: A review on early diagnosis performance

Chintaluri A[1], Radzihovsky M[2]

Submitted: September 29, 2024, Revised: version 1, December 23, 2024, version 2, December 29, 2024, version 3, January 22, 2025
Accepted: January 25, 2025

Abstract

Cardiovascular disease (CVD) poses a significant threat to global health, responsible for approximately one-third of all deaths worldwide. Early diagnosis plays a crucial role in preventing severe complications, yet traditional methods often require expensive tests and equipment. Machine learning offers an innovative and more cost-efficient alternative, through pattern recognition of comparatively lightweight data to accurately predict CVD presence and severity. In this paper, we augmented and improved conventional binary approaches to CVD diagnosis with quantitative ordinal labels, where a higher number represents more severity. Using the UC Irvine Heart Disease dataset, we tested six lightweight models — Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Support Vector Machines (SVC), and Decision Trees (DT) — alongside a custom neural network model to evaluate their performance on this multi-class classification task. Each model's accuracy, precision, recall, and F1 score were analyzed to assess robustness in clinical settings. The KNN, RF, and custom models achieved testing accuracies > 80%, with the KNN leading in accuracy, precision, recall, and F1-score, improving significantly over the LR model's accuracy of 53.2%. Our findings demonstrated that lightweight models are accurate for CVD prediction, offering potential for future deployment in AI-assisted tasks in hospitals. However, these models lack explainability behind the diagnosis, which is essential for real-world deployment. Future directions include exploring wearable technology for continuous CVD monitoring and the inclusion of worldwide data to account for risk factors in diverse populations.

Keywords

Cardiovascular disease, Machine learning, Random Forest, K-Nearest Neighbors, SMOTE, Skip connections, Deep learning, Wearable technology, Early diagnosis

[1]Corresponding Author: Anirudh Chintaluri, anirudh.chintaluri@gmail.com, Thomas Jefferson High School for Science and Technology, 6560 Braddock Rd, Alexandria, VA 22312, USA.
[2]Matthew Radzihovsky, radzihovsky@gmail.com, Experimental Atomic Physics Group, Massachusetts Institute of Technology, 50 Vassar St, Cambridge, MA 02139, USA.

Introduction

The heart, one of the most important and essential organs in the human body, pumps blood out and into the rest of the body to provide oxygen necessary for the body to function and stay alive. Since the heart is such an integral part of the human body, any disease that affects the cardiovascular system, commonly referred to a cardiovascular disease (CVD), puts life at risk, with complications resulting in long-term health conditions, as well as in fatality (1).

In the year 2022, according to the U.S. Centers for Disease Control and Prevention, over 700,000 deaths have been reported in the United States alone with one of many CVDs as the cause, including coronary artery disease, arrhythmia, and cardiac arrests (1). Furthermore, for the past 30 years, cases of CVD have been on the rise worldwide (2). When the COVID-19 pandemic spread worldwide in 2020 and 2021, CVD was identified as a major risk factor for severe complications from contracting the virus (2).

Historically, early detection of CVD has played a key role in preventing severe complications from emerging in patients, and has therefore resulted in more cost-effective healthcare to treat it (3). One of the emerging methods of detecting CVD at an early stage in particular is machine learning, which focuses on distinguishing and learning from patterns in CVD data (4).

Many previous studies have proposed machine learning techniques to accomplish this task of early detection. Some of these included support vector machines (SVC), multilayer perceptrons (MLP), decision trees (DT), and regression models such as logistic regression (LR) (4). Pal et al. used machine learning model types such as MLP and K-Nearest neighbor classifiers (KNN) to predict binary classification of CVD using data from the UC Irvine machine learning repository, with the MLP model achieving an accuracy score of 82.5% (5). Korial et al. while also using the same dataset, focused on an ensemble-based approach using models such as LR, KNN, RF, and a voting ensemble model for binary classification, and achieved an accuracy of 92.1% with their voting ensemble model (6).

Unlike many existing studies that focus primarily on binary classification, this paper focuses primarily on five different classifications depending on CVD severity, and our aim was to review many different types of supervised learning models, from ensemble models to deep learning models — with the latter including a custom-built neural network model that utilized the skip connection as data was fed into the network — using a variety of metrics, such as accuracy, F1 score, precision, and recall. The purpose of the diversity of metrics was to compare model robustness and relevant performance metrics pertaining to a clinical task such as CVD detection, as well as to identify what factors may predispose patients to CVD, and methods to mitigate its severity. Additionally, we intended to evaluate the effectiveness of machine learning in detecting CVD using features which could be measured by wearable technology. We hypothesized that machine learning models could effectively move beyond binary classification approaches, so as to be able to

accurately predict both the presence as well as the severity of CVD in patients.

Methods

Dataset and features

To obtain data for training, we utilized the Heart Disease dataset available to the public in the UC Irvine Machine Learning repository. The dataset contained 14 features pertaining to a patient and labeled the presence of heart disease, with 303 instances. Its attributes consisted of factors that either did not require tests or were easier to measure and cheaper than current CVD tests. These factors included age, sex, blood pressure, cholesterol, and electrocardiogram (ECG) results (9). The list of features are shown in Table 1.

Table 1. List of features from UC Irvine's heart disease dataset

Feature	Description
age	Quantitative measurement of the patient's age in years.
sex	Binary (0 = Female, 1 = Male).
cp	Chest pain type (1 = typical angina, 2 = atypical angina, 3 = non-anginal pain, 4 = asymptomatic)
trestbps	Resting blood pressure, in mmHg.
chol	Serum cholesterol, in mg/dl.
fbs	Binary, whether or not fasting blood sugar > 120 mg/dl (0-1).
restecg	Resting electrocardiographic results (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy).
thalach	Maximum heart rate achieved, in beats/min.
exang	Exercise induced angina (0 = No, 1 = Yes).
oldpeak	ST depression induced by exercise relative to rest.
slope	The slope of the peak exercise ST segment (1 = upsloping, 2 = flat, 3 = downsloping).
ca	Number of major vessels (0-3) colored by fluoroscopy.
thal	Thalassemia defect types: 3 = normal; 6 = fixed defect; 7 = reversible defect.
label	Categorical, ordinal (0-4). Signifies presence of heart disease of patient, with 0 indicating absence and 4 indicating most severe presence.

Preprocessing

All code that was written to preprocess and train our machine learning models was built on a Jupyter Notebook with Python 3.10. This platform better organized our code into smaller, individual cells for each step of model development.

The dataset has missing values. Therefore, we used a K-Nearest Neighbors (KNN) imputation approach to determine which missing value best aligned with the non-empty data. We used KNN imputation because it is considered as a more robust and better-performing imputation method as opposed to more naïve methods such as filling the empty data cells in using either

the mean, median, or mode of the data (8).

We normalized our data into either a 0-1 range or a z-score. Discrete values were normalized into the 0-1 range, and continuous values were normalized into the z-score range. Machine learning models more easily learn and converge when data is normalized within a specific range, due to the use of smaller numbers while also maintaining the distinction between categories and numbers. Normalization avoids bias of a particular feature, while also improving model outcomes when normalized by z-score and a 0-1 range (9).

Lastly, we sought to balance class distribution, since the negative case had about as many values as each of the four positive cases combined. To balance out the class distribution, we used the Synthetic Minority Over-Sampling Technique (SMOTE), an oversampling technique that synthetically creates data points from minority classes to match the number of samples from the majority class. As a result, SMOTE reduces bias towards the majority class and places a more even emphasis on the minority CVD classes (10). The result of using SMOTE was an equal distribution of all five labels, as shown in Figure 1. When sampling from a balanced dataset with 820 instances for the testing set, the test distribution is expected to remains approximately balanced as well; however, we still used a stratified split to confirm that our testing data was balanced (see later).

Tables 2 and 3 show the first 5 data points of the set before and after preprocessing the data respectively. Additionally, to address multicollinearity, we tested to determine whether or not two features were highly correlated to the point where they were rendered redundant. Highly correlated features can lead to misleading conclusions due to a decrease in the precision of the regression coefficients. Multicollinearity was determined using the Variance Inflation Factor (VIF). Features with VIF values greater than 6.0 were removed from the dataset, which in this dataset include cp, slope, and thal. An exception to this is cp, because it could be used by wearable technology as a user input to detect CVD. Results of the VIF test are shown in Figure 2.

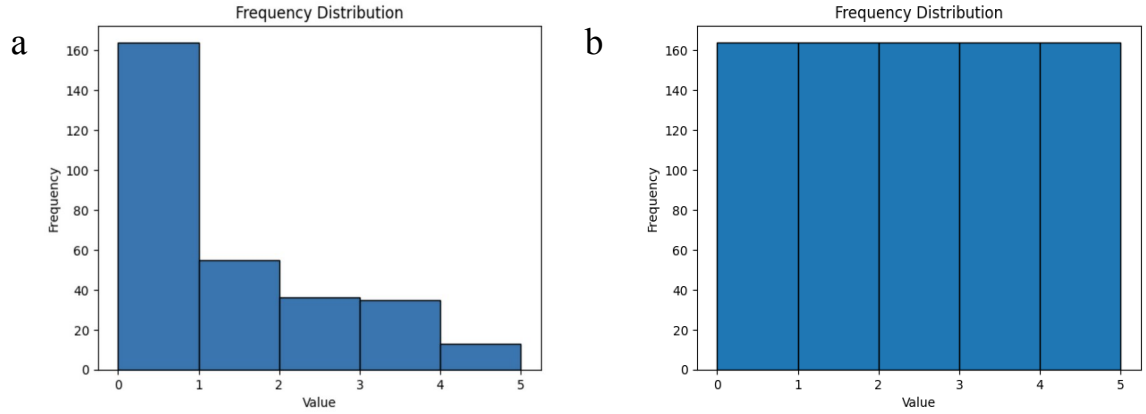

Figure 1. Frequency distributions of the class attribute (a) before resampling with SMOTE and (b) after resampling with SMOTE.

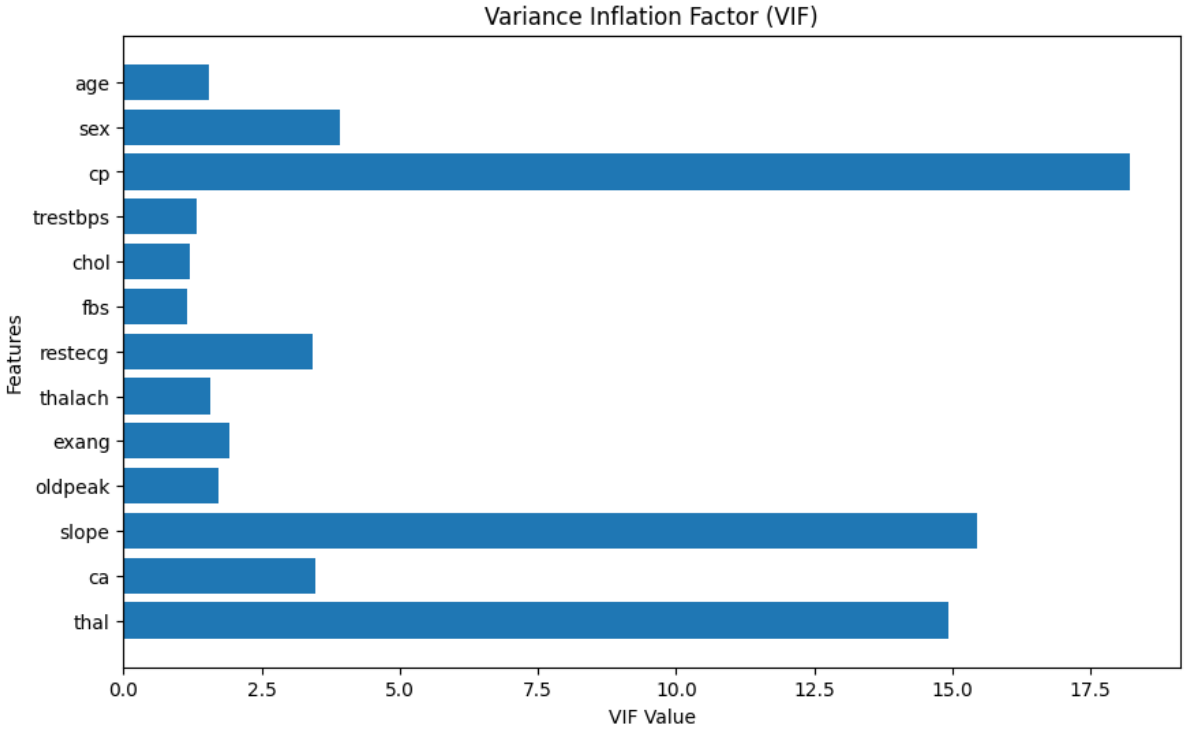

Figure 2. Results of the VIF test to detect multicollinearity

Table 2. List of the first 5 data points of the dataset before preprocessing

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	label
0	63	1	1	145	233	1	2	150	0	2.3	3	0.0	6.0	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3.0	3.0	2
2	67	1	4	120	229	0	2	129	1	2.6	2	2.0	7.0	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0.0	3.0	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0.0	3.0	0

Table 3. List of the first 5 data points of the dataset after preprocessing

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	ca	label
0	0.947160	1	0.25	0.756274	-0.264463	1	1.0	0.017169	0	1.085542	0.000000	0
1	1.389703	1	1.00	1.608559	0.759159	0	1.0	-1.818896	1	0.396526	1.000000	2
2	1.389703	1	1.00	-0.664201	-0.341717	0	1.0	-0.900864	1	1.343924	0.666667	1
3	-1.929372	1	0.75	-0.096011	0.063869	0	0.0	1.634655	0	2.119067	0.000000	0
4	-1.486829	0	0.50	-0.096011	-0.824558	0	1.0	0.978917	0	0.310399	0.000000	0

Model development

To analyze our model performance, the dataset was split randomly using stratified random sampling into two categories: 80% of the data for training, and 20% for testing, with the training set intended to train, and the testing dataset for an unbiased evaluation of model performance. Stratified sampling ensured an even distribution among the labels in the testing set.

The models used in this study included six different lightweight models in addition to a custom deep learning model. The six lightweight models are readily available in the scikit-learn library, and included the following: Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Support Vector Machines Classifier (SVC), and Decision Trees (DT). The hyperparameters used for each model are presented in Table 4.

Table 4. Hyperparameters for the models

Model	Hyperparameters
LR	Random seed 42
RF	128 trees, random seed 42
KNN	1 neighbor
MLP	Random seed 42
SVC	Random seed 42
DT	Random seed 42

The custom neural network model used in this study was built using Keras, made with 12 layers, 10 hidden layers, and 2 skip connections, for a total of 1,088,539 weights and biases. Skip connections in neural network models have been shown to improve performance by eliminating singularities, where node overlap and elimination can cause the model to acquire greater degeneracy (11). The model was trained using the Adam optimizer with a learning rate of 0.00001, the sparse categorical cross-entropy loss function, and with a batch size of 16 for a total of 400 epochs for training. This was performed by taking a small portion of the dataset (5%) and using it as validation data. The hyperparameters that yielded the best performance results were subsequently used.

Additionally, a callback was created in order to automatically save the best performing model during training into a .keras file. This performance callback depended on the performance of the validation dataset. Figure 3 displays the layout of the neural network.

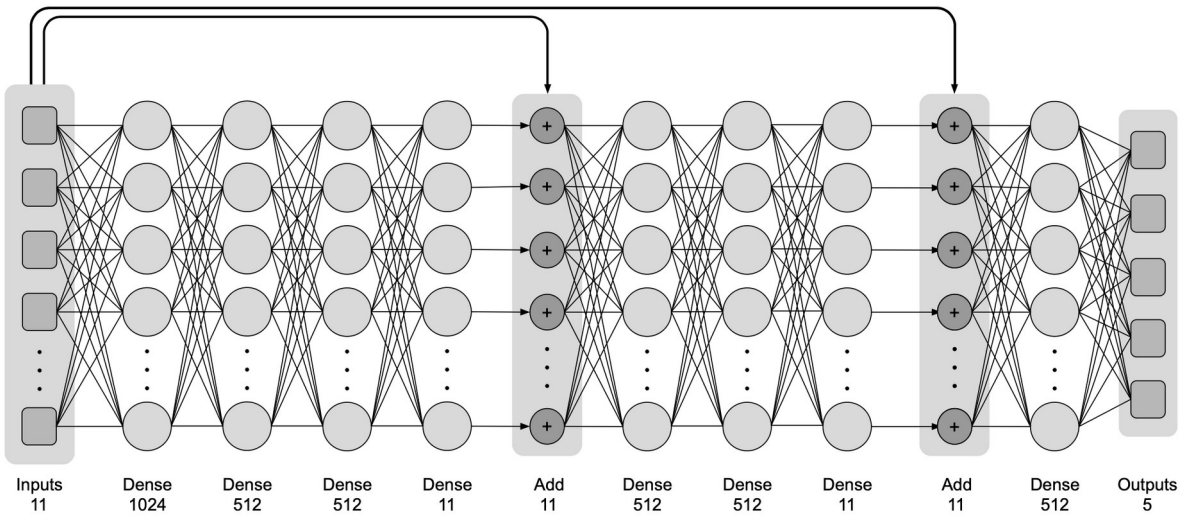

Figure 3. Layout of the custom neural network model tested

Data analysis

After testing the model, we compared the models using several different metrics, since accuracy does not provide a complete picture of how well a model performs. It is important, especially in clinical applications of machine learning, to minimize false negatives — that is, when a model predicts a negative value when it is in fact not. We also compared our custom model with the other six lightweight models with respect to metrics such as F1 score, recall, and precision, and averaged the scores across 5-fold cross-validation. The recall and precision metrics used in this study were calculated through macro-averaging, the arithmetic mean across all classes, to take into account all five label categories.

Results

Table 5 presents the performance of the seven different machine learning models tested on the UC Irvine Heart Disease Dataset using the testing set, with confusion matrices provided in Figure 4. Further details are shown in the Appendix section. The top performing models in this test are the custom Keras-built model, the RF, and the KNN, achieving testing accuracies of 0.834, 0.866, and 0.888 respectively. The KNN obtained the highest precision score with an average of 0.894, followed closely by the RF and the custom model, with scores of 0.869 and 0.839 respectively. Comparing the recall metric, the KNN performed best with a score of 0.888, followed by scores of 0.866 and 0.834 by the RF and the custom model respectively. The performance of these models with respect to the magnitudes of the precision, recall, and F1 scores further corroborated the acceptable accuracy of these models, obtaining scores very similar to the testing accuracy. The maximum standard deviation for accuracy, precision, and recall after cross-validation was 0.036, 0.044, and 0.036 respectively, attributable to the DT, LR, and DT models respectively. The custom model presented with the lowest standard

deviations, with scores of 0.007, 0.015, and 0.007 for the same respective metrics.

A recurring theme noticed when using these models was that even though some of them presented with relatively lower testing accuracies, their AUC values were disproportionately higher. Though the custom and RF models did not score as high as the KNN in terms of accuracy, they achieved AUC scores of 0.981 and 0.955 respectively, higher than the 0.93 of the KNN. In fact, even the MLP, with an accuracy of 0.771, still presented with a higher AUC than the KNN with a score of 0.937. This could suggest that the classifier achieves the good performance on the positive class (high AUC), at the cost of a high false negative rate.

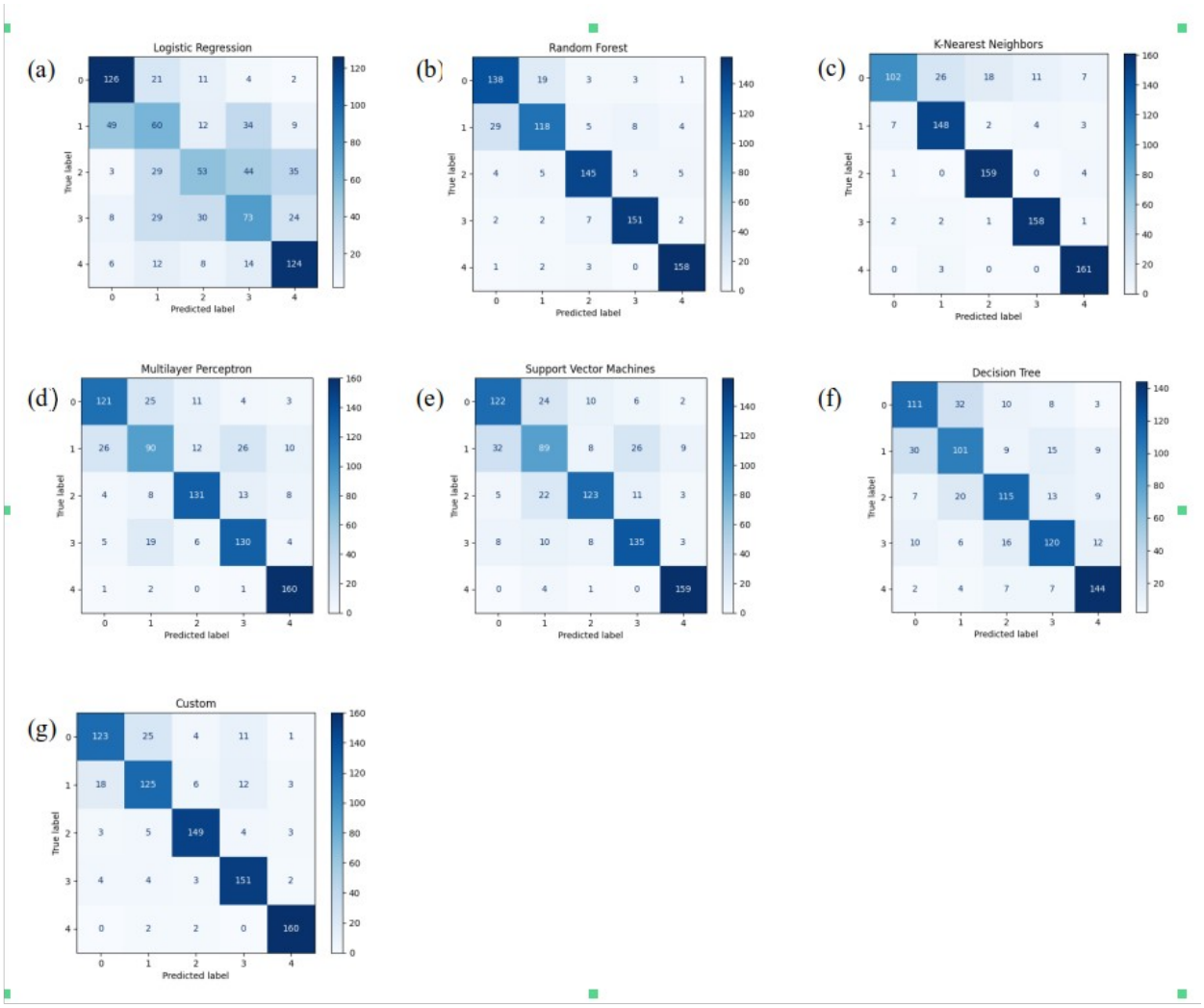

Figure 4. Aggregate confusion matrices of all 7 models, added up from 5-fold cross-validation

Table 5. Comparison of seven different machine learning models in average testing accuracy, precision, recall, and F1 score. Numbers include mean ± standard deviation

Model Type	Testing Accuracy	Precision	Recall	F1 Score	AUC
LR	0.532 ± 0.02	0.523 ± 0.029	0.532 ± 0.02	0.527 ± 0.027	0.819 ± 0.013
RF	0.866 ± 0.025	0.869 ± 0.024	0.866 ± 0.026	0.867 ± 0.028	**0.981 ± 0.003**
KNN	**0.888 ± 0.013**	**0.894 ± 0.015**	**0.888 ± 0.012**	**0.891 ± 0.015**	0.93 ± 0.008
MLP	0.771 ± 0.036	0.767 ± 0.044	0.771 ± 0.036	0.769 ± 0.045	0.937 ± 0.012
SVC	0.766 ± 0.025	0.763 ± 0.027	0.766 ± 0.025	0.768 ± 0.031	0.93 ± 0.01
DT	0.721 ± 0.021	0.725 ± 0.018	0.721 ± 0.021	0.723 ± 0.022	0.825 ± 0.013
Custom	0.834 ± 0.007	0.839 ± 0.015	0.834 ± 0.007	0.837 ± 0.012	0.955 ± 0.011

Discussion

The purpose of this study was to provide a review of machine learning efficacy in identifying risk factors for, and accurately diagnosing cardiovascular disease in patients. The results of this study demonstrated the potential of machine learning models in accurately classifying the severity of CVD in patients. For the RF, KNN, and custom models in particular — with accuracy scores above 80% for each — the results demonstrated a significant improvement over the baseline LR model, which yielded an average testing accuracy score of 53.2%, as well as the benchmark standard of 20% for a random model that (randomly) predicted one of five categories. This suggested that lightweight models such as RF and KNN could accurately classify the severity of CVD from easily measurable and minimally onerous biomarkers or anatomical/physiological attributes. Additionally, the inclusion of other metrics, including precision, recall, F1 score, and AUC, demonstrated that the features chosen were suited to machine learning approaches for this clinical task, when minimizing false negatives may be of primary concern.

Our findings align with prior research in that machine learning can be effective and accurate in detecting cardiovascular disease. However, our approach extrapolated beyond the binary classification used in prior studies, and instead focused on the spectrum of severity, making possible a more nuanced review of the patient's condition. As a result, medicines could be prescribed that are (mechanism and dose) appropriate for that stage in the severity and progression of the disease; as can the frequency of follow-up visitation and care.

Despite the robustness and acceptable accuracy of these models, they are not ready for full self-deployment in hospitals. For a clinical-based machine learning model to truly provide useful information about a patient's medical diagnosis, and to be approved by regulatory agencies, it must be able to reason and explain why that patient is predisposed to, or is at risk of, or has CVD (12). It is not acceptable for a deployed model to not provide an explanation as to what exactly led to the positive test of CVD, and in more specific scenarios, the severity of the CVD. The inability to access and/or measure data quality, bias and

computational complexity/robustness of commercial (often) proprietary models affects their explainability and interpretability, turning them into 'black boxes', unsuitable for regulatory approval. At this point in time, models like this instead can only serve as an AI-assisted tool to avoid bias and provide an objective prediction rather than as a fully independent diagnosis tool.

In the future, a direction worth exploring is how these models can be incorporated in wearable technology. With existing equipment in hospitals, a medical diagnosis is still expensive – even with insurance. As wearable technology such as smartwatches becomes more prevalent in society, there is potential for what is already an active health monitor to monitor risk of mild or even severe cardiovascular disease. Furthermore, with active monitoring of a patient's data on a smartwatch, it may be even more capable of responding to early signs of cardiovascular disease, preventing further complications and severe symptoms from emerging in the patient, and these can be lifesaving. A subset of the existing CVD dataset can be used to accommodate wearable technology, removing all but the age, sex, cp, trestbps, restecg, and thalach attributes, as these attributes are most easily accessible without a doctor's visit. An analysis of feature importance was summarized in Figure 5 utilizing a Pearson and Gini feature importance analysis, showing that age, trestbps, and thalach were the three most important features to examine from this dataset in terms of wearable tech. The Pearson Correlation was based on measuring correlation between each feature and the label, and the Gini Importance was derived from the Random Forest model, one of the higher-performing models. With correlation values as extreme as 0.41 for cp and -0.42 for thalach, it seems to be quite plausible to inform the wearer when calculated severity triggers alarms to engage in immediate evasive action. However, the high multicollinearity with cp suggests that it may not be nearly as important to predict the label, as reflected with the low Gini score of 0.15.

In practice, however, the opposite proved to be true. The best-performing machine learning model used on the original dataset, the KNN model, achieved an accuracy of 0.778 across 5-fold cross-validation with the wearable version of this dataset, being outperformed by all but the LR and DT models from the original dataset. An accuracy of 0.778 is not sufficient given the context of clinical testing due to the aforementioned reasons. In addition, there is a very high ~ one-in-four chance that the model misdiagnoses the patient.

Another future direction includes obtaining data from patients all over the world. In this study, we discussed relationships among several factors in Cleveland alone; however, there may be other factors not mentioned in this study that are important to cardiovascular disease diagnoses, such as air quality, diet and immunity; among others. This can provide us with an even more nuanced review of the most important risk factors predisposing to, or increasing the risk of CVD.

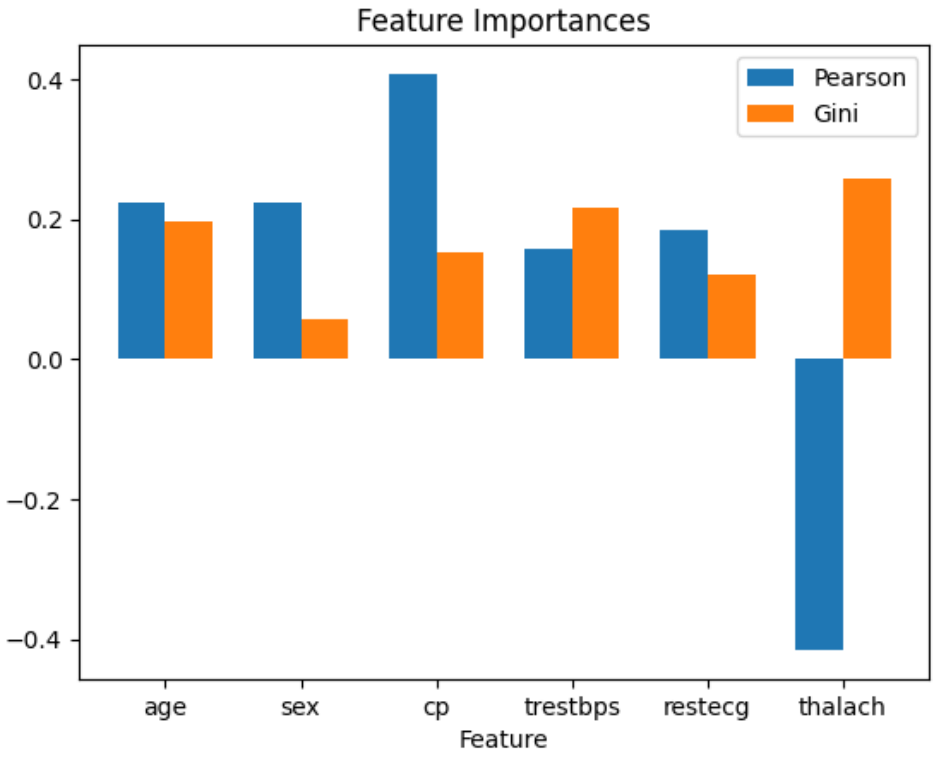

Figure 5. Pearson correlations and Gini importance of attributes, tested with the adjusted wearable dataset

Conclusion

The study demonstrated that machine learning can be effective at both saving the costs of expensive tests, and preserving and prolonging patient quality of life with its ability to detect CVD early. We determined the best ML model which could predict the severity of CVD based on features obtained from a publicly available dataset. The ability to accurately predict severity on a sufficiently nuanced and differentiable scale of 0 through 4; as opposed to simply a binary determination of the presence of absence of CVD increases the actionable time period for the appropriate medication type and dose and lifestyle course corrective measures. The lightweight models such as KNN and RF, as well as our deep custom neural network model, stood out in that regard, producing accuracy, precision and recall metrics of > 0.83 and AUC of > 0.93. Furthermore, F1 values of > 0.83 indicated that the constructed models were robust, where precision was not optimized at the expense of recall; or vice' versa. Such traits are particularly useful within the context of clinical diagnosis. Though these models do not fulfill the conditions for standalone real-world deployment, they mark a step forward with machine learning technology and can be used for AI-assisted CVD diagnosis.

Data repositories

All code used for the development of our machine learning models, including steps

towards building them, is publicly available in the following GitHub repository: https://github.com/anirudhc5/cvd-machine-learning-notebook. The purpose of random seeds in this study is for the reproducibility of data. The dataset used for building this model is available to the public on the UC Irvine Machine Learning Repository: https://doi.org/10.24432/C52P4X.

Abbreviations

CVD: Cardiovascular Disease, LR: Logistic Regression, RF: Random Forest, KNN: K-Nearest Neighbors, MLP: Multilayer Perceptron, SVC: Support Vector Machines, DT: Decision Trees

Acknowledgement

We would like to express our appreciation to the Inspirit AI 1:1 Mentorship Program for their time investment and making this work possible.

References

1. "Heart Disease." U.S. Centers for Disease Control and Prevention, 15 may 2024. www.cdc.gov/heart-disease/about/index.html#:~:text=What%20is%20heart%20disease%3F,can%20cause%20a%20heart%20attack

2. "Cardiovascular Disease Is on the Rise, but We Know How to Curb It. We've Done It before." National Heart, Lung, and Blood Institute, 3 Feb. 2021. www.nhlbi.nih.gov/news/2021/cardiovascular-disease-rise-we-know-how-curb-it-weve-done-it#:~:text=Over%20the%20last%2030%20years,across%20the%20globe%20external%20link%20

3. Wolcherink MJO, Behr CM, Pouwels XGLV, Doggen CJM, Koffijberg H. "Health Economic Research Assessing the Value of Early Detection of Cardiovascular Disease: A Systematic Review." PharmacoEconomics, 41(10), 1183-1203, 2023. https://doi.org/10.1007/s40273-023-01287-2

4. Latha CBC, Jeeva SC. "Improving the Accuracy of Prediction of Heart Disease Risk Based on Ensemble Classification Techniques." Informatics in Medicine Unlocked,16, 100203, 2019. https://doi.org/10.1016/j.imu.2019.100203

5. Pal M, Parija S, Panda G, Dhama K, Mohapatra RK. "Risk prediction of cardiovascular disease using machine learning classifiers" Open Medicine, 17(1), 1100-1113, 2022. https://doi.org/10.1515/med-2022-0508

6. Korial AE, Gorial II, Humaidi AJ. "An Improved Ensemble-Based Cardiovascular Disease Detection System with Chi-Square Feature Selection." Computers, 13(6),126, 2024. https://doi.org/10.3390/computers13060126

7. Janosi A, Steinbrunn W, Pfisterer M, Detrano R. "Heart Disease." [Dataset], UCI Machine Learning Repository, 1989. https://doi.org/10.24432/C52P4X.

8. Jadhav A, Pramod D, and Ramanathan K. 'Comparison of Performance of Data Imputation Methods for Numeric Dataset'. Applied Artificial Intelligence, 33(10), 913–933, 2019. https://doi.org/10.1080/08839514.2019.1637138.

9. Singh D, Singh B. "Investigating the Impact of Data Normalization on Classification Performance." Applied Soft Computing, 97, 2019. http://dx.doi.org/10.1016/j.asoc.2019.105524

10. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research, 16, 321-57, 2002. https://doi.org/10.1613/jair.953

11. Orhan AE, Pitkow X. "Skip Connections Eliminate Singularities." arXiv preprint, 2017. https://doi.org/10.48550/arXiv.1701.09175

12. Kononenko I. "Machine Learning for Medical Diagnosis: History, State of the Art and Perspective." Artificial Intelligence in Medicine, 23(1), 89-109, 2001. https://doi.org/10.1016/s0933-3657(01)00077-x.

Appendix

Table A1. Performance metrics of machine learning models across all five folds.

Model	F1-Score	Accuracy	Precision	Recall	AUC
Logistic Regression	0.519318425	0.518292683	0.519129411	0.519507576	0.803842817
	0.535958112	0.548780488	0.524684087	0.547727273	0.827656214
	0.485375721	0.5	0.471920062	0.499621212	0.803444873
	0.554142596	0.554878049	0.554878049	0.553409091	0.831958059
	0.541626712	0.536585366	0.545038292	0.538257576	0.82751453
Random Forest	0.892186347	0.890243902	0.893657834	0.890719697	0.98088314
	0.8483079	0.847560976	0.848699495	0.847916667	0.980432606
	0.831479304	0.829268293	0.835342664	0.827651515	0.978425175
	0.868147611	0.865853659	0.870589181	0.865719697	0.978739627
	0.896794408	0.896341463	0.897377451	0.896212121	0.987014407
K-Nearest Neighbors	0.901085493	0.896341463	0.893657834	0.896969697	0.935513532
	0.882898564	0.884146341	0.848699495	0.884469697	0.927748381
	0.895277359	0.890243902	0.835342664	0.890340909	0.931453129

	0.869118497	0.865853659	0.870589181	0.866287879	0.916361612
	0.905723281	0.902439024	0.897377451	0.901325758	0.938460704
Multilayer Perceptron	0.756801893	0.756097561	0.756029608	0.757575758	0.927497542
	0.794957873	0.792682927	0.797316017	0.792613636	0.944604441
	0.695092448	0.707317073	0.685531136	0.704924242	0.91745135
	0.806077021	0.804878049	0.807233106	0.804924242	0.94577225
	0.791197369	0.792682927	0.789410423	0.792992424	0.949745186
Support Vector Machines	0.753543689	0.75	0.755583208	0.751515152	0.921345492
	0.789733928	0.792682927	0.786502101	0.792992424	0.936383154
	0.718751828	0.725609756	0.713711712	0.723863636	0.915723311
	0.789482051	0.774390244	0.772834033	0.774621212	0.938931298
	0.787306833	0.786585366	0.787872052	0.786742424	0.937863607
Decision Tree	0.739470945	0.737804878	0.74011726	0.738825758	0.836623149
	0.750577567	0.75	0.751346099	0.749810606	0.843653857
	0.717451656	0.719512195	0.716722976	0.718181818	0.824028452
	0.700083321	0.695121951	0.705163561	0.695075758	0.809416204
	0.705411099	0.701219512	0.709544315	0.701325758	0.813316129
Custom	0.852695817	0.841463387	0.863419913	0.842234848	0.960721721
	0.822589515	0.823170722	0.821696127	0.823484848	0.958808625
	0.828244732	0.829268277	0.828080439	0.828409091	0.952191765
	0.834982446	0.835365832	0.834359762	0.835606061	0.936042679
	0.844955102	0.841463387	0.848461799	0.841477273	0.968763915