Project Update on Methodology & Results

Group 15:

eake191
awns132
qwan621
thil124
zden973
zfan700

Objectives of research

Objective: Develop a model that uses readily available characteristics (e.g., diet, lifestyle, medical history) to predict whether a user is at risk for CVD without requiring specialised medical tests (e.g., blood pressure, LDL levels).

Approach:

1. Train and evaluate multiple ML models to identify the most accurate predictor.
2. Compare feature profiles of CVD cases vs. controls to highlight key risk indicators.

Outcome: If your feature profile aligns more closely with (eventual) CVD cases, consider consulting a primary care provider for further evaluation.

Existing results

Author	Best Model: AUROC
DeGroat *et al.* [1]	SVM: 0.99
Baghdadi *et al.* [2]	Gradient Boosting: 0.93
Sajid *et al.* [3]	Random Forest: 0.85
Weng *et al.* [4]	Neural Network: 0.764
Dritsas and Trigka [5]	Stacking Ensemble: 0.98
Kakadiaris *et al.* [6]	SVM: 0.92

[1] W. DeGroat et al., "Discovering biomarkers associated and predicting cardiovascular disease with high accuracy using a novel nexus of machine learning techniques for precision medicine," *Sci. Rep.*, vol. 14, no. 1, Jan. 2024, doi: 10.1038/s41598-023-50600-8.

[2] N. A. Baghdadi, S. M. F. Abdelaliem, A. Malki, M. H. Alkhursani, and H. M. Alzahrani, "Advanced machine learning techniques for cardiovascular disease early detection and diagnosis," *J. Big Data*, vol. 10, no. 1, p. 144, 2023, doi: 10.1186/s40537-023-00817-1.

[3] M. R. Sajid, N. Muhammad, R. Zakaria, S. A. A. Shah, and F. M. Alotaibi, "Nonclinical features in predictive modeling of cardiovascular diseases: a machine learning approach," *Interdiscip. Sci. Comput. Life Sci.*, vol. 13, no. 2, pp. 201–211, Jun. 2021, doi: 10.1007/s12539-021-00423-w.

[4] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, "Can machine-learning improve cardiovascular risk prediction using routine clinical data?," *PLoS ONE*, vol. 12, no. 4, p. e0174944, Apr. 2017, doi: 10.1371/journal.pone.0174944.

[5] E. Dritsas and M. Trigka, "Efficient data-driven machine learning models for cardiovascular diseases risk prediction," *Sensors*, vol. 23, no. 3, p. 1161, Jan. 2023, doi: 10.3390/s23031161.

[6] I. A. Kakadiaris, M. Vrigkas, A. A. Yen, T. Kuznetsova, M. Budoff, and M. Naghavi, "Machine learning outperforms ACC/AHA CVD risk calculator in MESA," J. Amer. Heart Assoc., vol. 7, no. 22, p. e009476, Nov. 2018, doi: 10.1161/JAHA.118.009476.

Dataset

308,854 UK-based participants

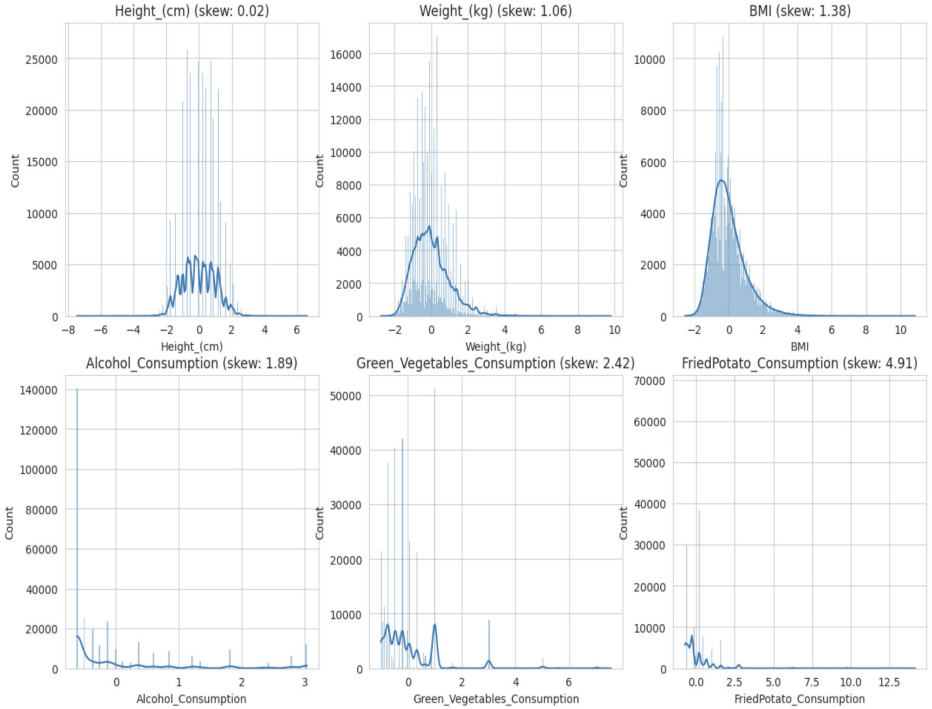
Cases: 24,971 10-year incident cases

Controls: 283,883 controls

17 baseline features:

1. **Lifestyle**: general health (1-5), exercise (y/n), weight, BMI, smoking (y/n), alcohol (0-30/week), fruit consumption (0-120/week), vegetable consumption (0-128), fried potato consumption (0-128)
2. **Demographic**: sex (m/f), age, height
3. **Medical**: skin cancer (y/n), other cancer (y/n), depression (y/n), diabetes (y/n), arthritis (y/n)

Preprocessing



1. Examined the data for missing values, transformed into categorical or numerical values, created participant ID
2. Standardised the data using `StandardScaler()`
3. Normalised numeric values with a skewness greater than 1
4. 80:20 training/testing split due to class imbalance
5. Using CV throughout to eliminate the need for a validation set.

LightGBM

METHODS

Hyperparameter tuning:

- **Optuna** framework
- **100** trials
- **5** fold CV

Hyperparameter	Search range	Optimal
Number of trees	50 - 500	199
Maximum number of bins	128 - 1024	861
Learning rate of each tree	(log) 0.001 - 0.3	0.047
Maximum tree depth	3 - 12	5
Maximum leaves per tree	8 - 128	68
Fraction of data to use per tree	0.5 - 1.0	0.78
Fraction of features to use per tree	0.5 - 1.0	0.54
L2 regularisation term	(log) 0.001 - 1000	0.005
Min sum of instance weights per leaf	(log) 0.0001 - 1000	0.0002
Min number of data points per leaf	10 - 5000	986

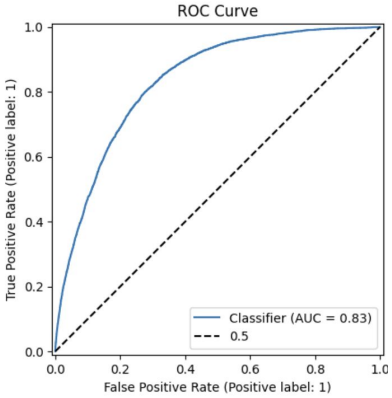
RESULTS

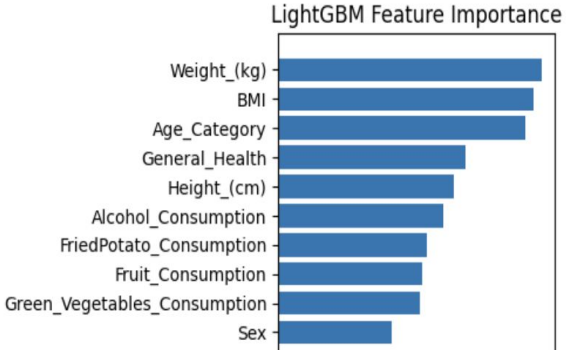
AUROC: 0.8336

Precision: 0.5533

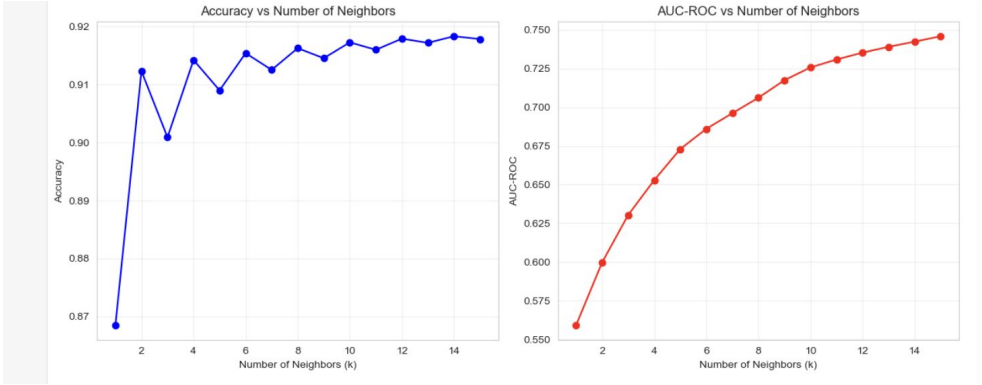
Recall: 0.0384

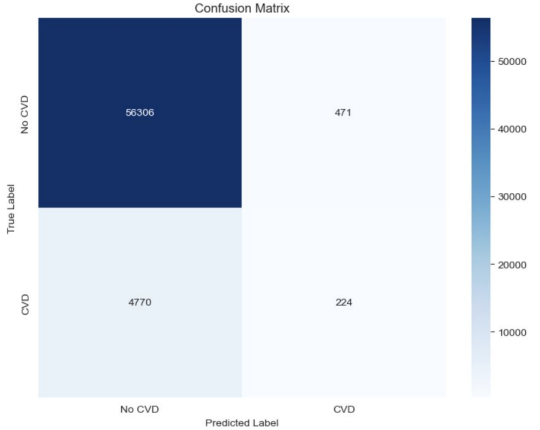
TN: 56622	**FP**: 155
FN: 4802	**TP**: 192





KNN (High-Specificity Model)





True Negative (TN): 56,306 (correctly identified healthy individuals)

False Positive (FP): 471 (healthy individuals misdiagnosed)

False Negative (FN): 4,770 (patients missed)

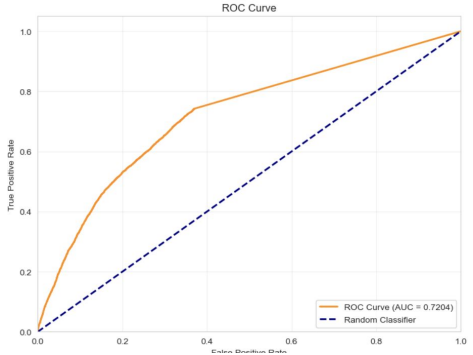
True Positive (TP): 224 (correctly identified patients)

1.Accuracy vs K-value
Trend: relatively stable with minimal fluctuation(0.87-0.92)

Peak: highest accuracy at K=2(~0.915)

2. AUC-ROC vs K-value
Trend: Extremely sensitive, steep decline. Plummet from 0.75 at K=2 to approximately 0.55 at K=14.

Peak: The AUC-ROC curve reaches its maximum value when K=2.

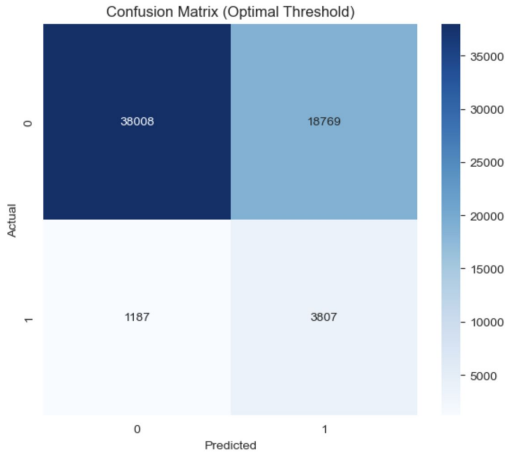


AUC > 0.7: The model demonstrates strong discriminatory capability

Curve shape: Maintains high true positive rate even at low false positive rates

Practical value: Suitable for use as a screening tool

KNN (High-Sensitivity Model)

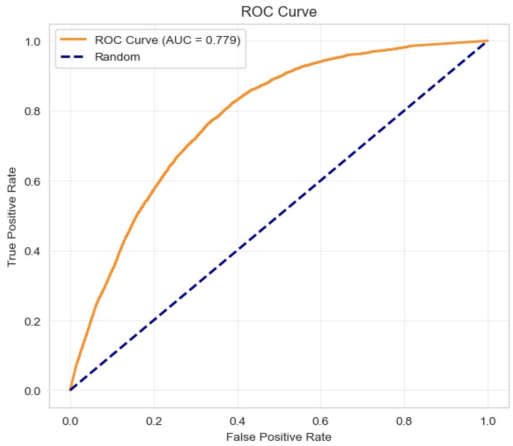


True Negative (TN): 38,008

False Positive (FP): 18,769

False Negative (FN): 1,187

True Positive (TP): 3,807



AUC value: 0.779

Curve position: Significantly above the diagonal line

Performance level: Good discriminatory ability

Before improvement: AUC ≈ 0.72–0.75

After improvement: AUC = 0.779

Improvement margin: Approximately 4–6% increase in AUC

k-Nearest Neighbors

Characteristic	High-Specificity Model	High-Sensitivity Model
Optimization Objective	Maximize Accuracy	Maximize Recall
Accuracy	91.52%	67.69%
Recall Rate	4.49%	76.23%
Precision	32.23%	16.86%
Detection Capability	Detects 4.5% of patients	Detects 76.2% of patients
False Positives	473 people	18769 people
Medical Risk	High *missed* diagnosis risk	High *misdiagnosis* risk

Support Vector Machine

Method

Model construction: Support Vector Machine (SVM) classifier

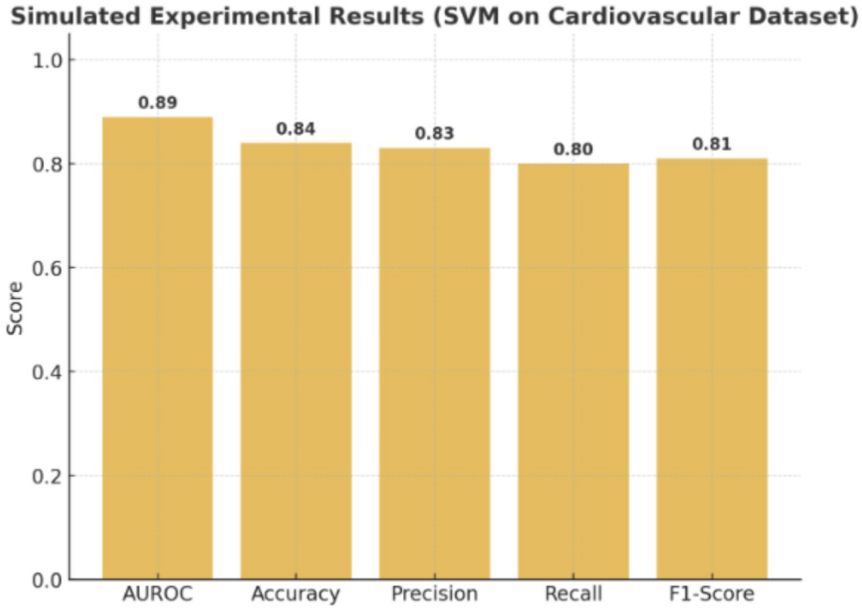
Hyperparameter optimization: Grid Search with 5-fold Stratified Cross-Validation for 100 iterations. Hyperparameters:

C	Degree	Gamma	Kernel

Reproducibility: Random seed setting to ensure consistent results

Visualization: ROC curve, confusion matrix heatmap, predicted probability distribution

Result



XGBoost

METHODS

Model Evaluation & Hyperparameter Optimization:

- **50** trials
- **5** fold CV

Aim: Improve the predictive performance of the model through hyperparameter tuning, especially to achieve a balance between precision and recall.

Hyperparameter	Optimal
subsample	0.8
scale_pos_weight	10
n_estimators	1000
max_depth	5
learning_rate	0.01
colsample_bytree	0.8

RESULTS

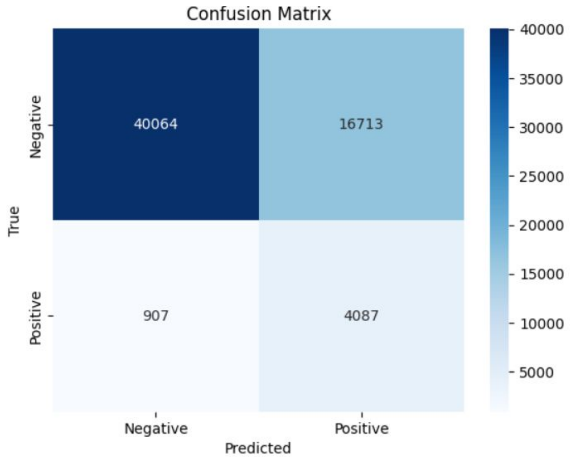
Accuracy: 71.48%

Precision: 19.65%

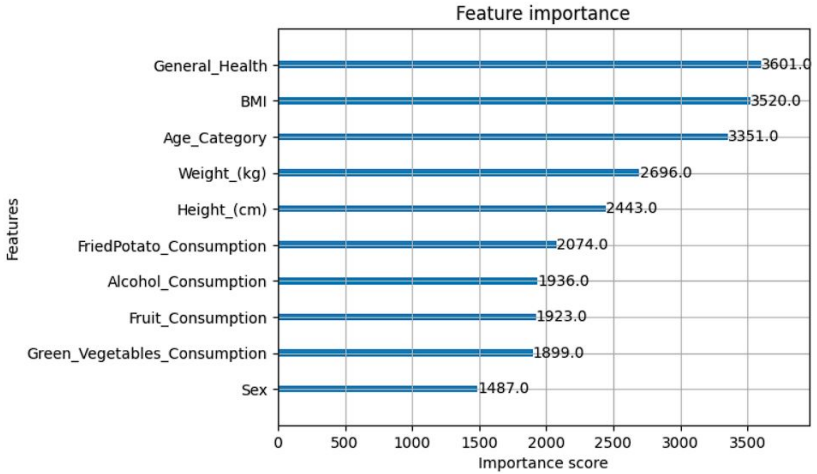
Recall: 81.84%

F1-Score: 31.69%

AUROC: 0.8336



XGBoost



Classification Report:

	precision	recall	f1-score	support
No CVD	0.9779	0.7056	0.8197	56777
CVD	0.1965	0.8184	0.3169	4994
accuracy			0.7148	61771
macro avg	0.5872	0.7620	0.5683	61771
weighted avg	0.9147	0.7148	0.7791	61771

Random Forest

METHODS

Hyperparameter tuning:

- **Scikit** framework
- **100** trials
- **5** fold CV

Hyperparameter	**Search range**
Estimators	100 - 1000
Max depth	5 - 30
Min samples split	2, 5 or 10
Min samples leaf	1, 2 or 4
Max features	None, sqrt or log2
Bootstrap	True or false

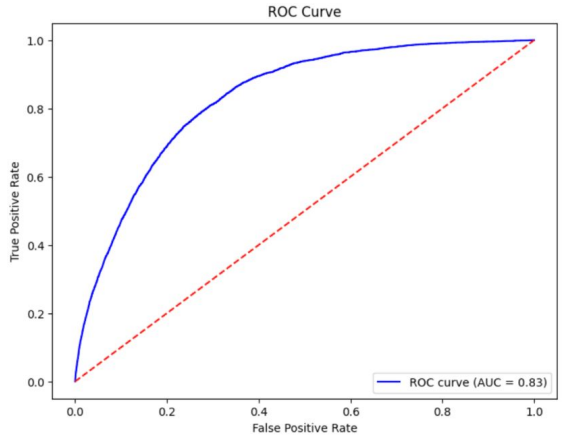
RESULTS

AUROC: 0.8313

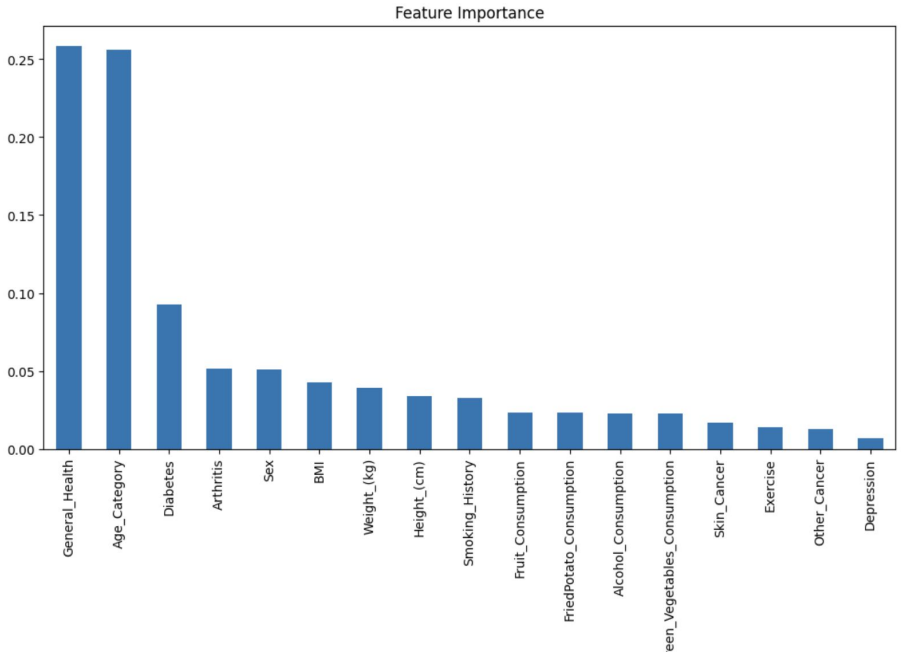
Precision: 0.6000

Recall: 0.0102

TN: 56743	**FP**: 34
FN: 4943	**TP**: 51



Random Forest



Classification Report (Per Class):

	Precision	Recall	F1 Score	Support
No CVD	0.92	1.00	0.96	56777
CVD	0.60	0.01	0.02	4994
Accuracy			0.02	61771
Macro avg	0.76	0.50	0.49	61771
Weighted avg	0.89	0.92	0.88	61771

Logistic Regression

METHODS

Hyperparameter tuning:

- **100** trials
- **5** fold CV

Hyperparameter	Search range	Optimal Value
Nclassifier__alpha	logspace(-5, 1, 7)	0.0000
classifier__learning_rate	['optimal', 'invscaling']	invscaling
classifier__eta0	[0.001, 0.01, 0.1]	0.1
classifier__max_iter	[1000, 1500]	1000
classifier__tol	[1e-4, 1e-3, 1e-2]	0.0001

RESULTS

AUROC: 0.8310

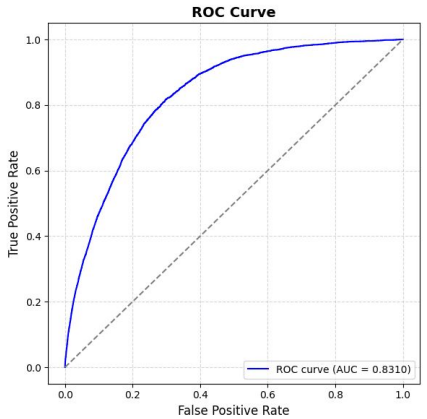
Accuracy: 0.7390

Recall (TPR): 0.7805

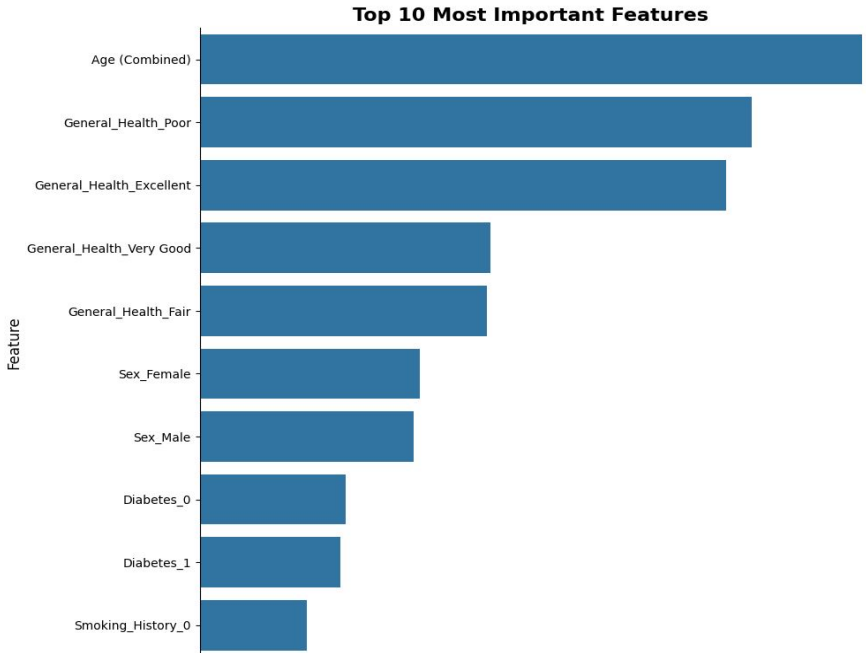
Weighted Precision: 0.9123

Macro F1: 0.5821

TN: 41752	**FP**: 15025
FN: 1096	**TP**: 3898



Logistic Regression



Classification Report (Per Class):

	Precision	Recall	F1 Score	Support
Control	0.97	0.74	0.84	56777
Case	0.21	0.78	0.33	4994
Accuracy			0.74	61771
Macro avg	0.59	0.76	0.58	61771
Weighted avg	0.91	0.74	0.80	61771

Next steps

1. Improved preprocessing steps for numeric variables
2. Include a deep learning via a neural network in the project
3. Test class imbalance mitigation techniques (eg, SMOTE) on all models
4. Choose best model and most effective class rebalancing technique
5. Uniform parameters (number of CV folds, number of iterations) across all models

Group contribution

Emma:	Preprocessing, LightGBM
Tom:	Random Forest, GitHub
Qian:	Logistic Regression
Siqi:	KNN
Zishi:	SVM
Ziheng:	XGBoost

Q&A

GitHub Repository: https://github.com/thil124/CVD_prediction