Given Personal Features about Individuals, Compare Supervised Learning Techniques to Predict Whether a Patient has a Heart Disease

Final Project Report

Javier GONZALEZ
Peter KESZTHELYI
Oscar MEURER
Alix VERMEULEN

Abstract

This paper gives an overview of various supervised machine learning techniques for the binary prediction task of whether a patient has a heart disease or not. Simpler techniques such as logistic regression performed poorly on our data as they would only predict one category of patients (healthy ones). The best results came from training a neural network with 5 dense layers and 400,417 trainable parameters which correctly predicted 91% of the patients with heart disease from the test set at the cost of predicting 42% of the healthy ones as sick unfortunately. A trade-off was apparent between the recall of unhealthy patients and the precision of healthy patients. Considerations were also made with the computing processing power required for certain models.

1. Introduction

Heart disease has been the **leading cause of death** in the United States since 1950. Risk factors for heart disease include high blood pressure, high cholesterol, smoking, diabetes, overweight and obesity, unhealthy diet, physical inactivity, and excessive alcohol use. In 2019, there was an average of 161,5 deaths per 100,000 people in the United States. Although the numbers are decreasing, it is still among the major causes of death.

According to health researchers, a key factor to lower the likelihood of dying from heart disease is **early stage diagnosis**. This way, the patient could begin actions to decrease the advancement of the disease, and moreover, improve his/her quality of life.

A potential application for the heart disease predictor is to **provide information to people as to whether they could have a heart disease or not**. This will make patients more self-aware and allow doctors to perform more robust and specific tests on high-risk patients and therefore decrease the number of people who die from heart diseases.

2. Problem Definition

The problem faced is a binary classification problem. The aim is to compare simple machine learning models such as Logistic Regression and gradually build to advanced techniques such as ensemble and deep learning in order to find the best predictor of heart disease. Category 0 patients will be “healthy” patients, unlikely to have a heart disease and category 1 patients will be “unhealthy” patients, likely to have a heart disease. Our goal is to maximise recall of category 1 patients while simultaneously minimising the false positives (predicting a healthy patient as unhealthy). We have decided to have thresholds of 85% for the recall of category 1 and 98% for the precision for category 0.

3. Related Work

There is plenty of work related to the field of heart disease prediction using machine learning methods.

When using decision trees and random forests, (Abdullah & Rajalaxmi, 2012, 22-25), used the Cleveland dataset consisting of 13 attributes. The results of this paper showed that the random forest model obtained a global accuracy of 63% and RMS error of 0.313.

Related to the use of neural networks, (AL-MILLI, 2013), based on the Cleveland dataset consisting of 13 attributes, trained a back propagation neural network to classify a patient into 4 categories. 83% global accuracy was achieved.

(El-Hasnony et al., 2022) also used the Cleveland dataset with 13 features to train a label ranking classifier. As a result, it achieved an accuracy and F-score of 57.4 ± 4% and 62.2 ± 3.6%, respectively.

4. Methodology

The work process is represented in the flowchart below: a first exploratory phase to analyse the data and correlation of features. After, a data pre-processing setup according to the results found in the data exploration phase. Then, a sampling method phase is set to evaluate the best sampling technique in terms of model performance and computer resources required. This phase was almost parallel to the exploration of machine learning models, where various basic models such as Naive Bayes, Logistic Regression, SVM, Decision Trees and KNN (K-Nearest Neighbour) were tested. Finally, ensemble and deep learning techniques were tested and tuned.

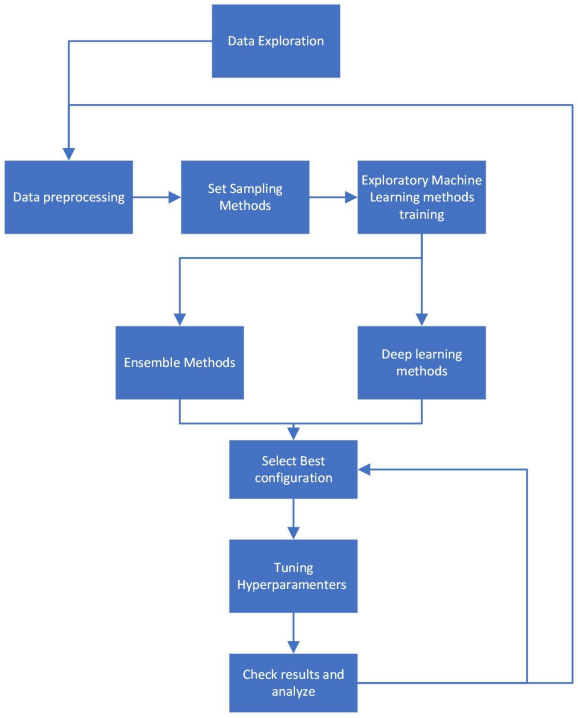

Fig 1. Methodology diagram.

The data was collected from Kaggle's "Personal Key Indicators of Heart Disease". This dataset has 400 000 instances and 18 features. The 18 features and their description are below:

Feature Name	Description
HeartDisease	Respondents reported having coronary heart disease (CHD) or myocardial infarction (MI)
BMI	Body Mass Index
Smoking	Respondents who have smoked ≥100 cigarettes in their life? [5 packs = 100 cigarettes]
AlcoholDrinking	Heavy drinkers (adult men having >14 drinks/week; adult women having >7 drinks/week)
Stroke	(Ever told) (you had) a stroke?
PhysicalHealth	For how many days during the past 30 days was your physical health not good?
MentalHealth	For how many days during the past 30 days was your mental health not good?
DiffWalking	Do you have serious difficulty walking or climbing stairs?
Sex	Are you male or female?
AgeCategory	Fourteen-level age category
Race	Race/ethnicity value
Diabetic	(Ever told) (you had) diabetes?
PhysicalActivity	Done physical activity or exercise during the past 30 days other than their regular job?
GenHealth	Would you say that in general your health is [Excellent, Very good, Good, Fair, Poor]
SleepTime	On average, how many hours of sleep do you get in a 24-hour period?
Asthma	(Ever told) (you had) asthma?
KidneyDisease	Not including kidney stones, bladder infection or incontinence, were you ever told you had kidney disease?
SkinCancer	(Ever told) (you had) skin cancer?

After performing Exploratory Data Analysis (EDA) on the data, we realised that the data was heavily imbalanced. Over 91% of patients didn't have a heart disease and close to 9% did have a heart disease. After removing duplicates from the dataset, this led to a 10,7:1 ratio of category 0 to category 1 labels. We therefore had to use sampling techniques to overcome this. We tried: random oversampling, random undersampling, NearMiss and SMOTE. SMOTE (Chawla et al., n.d., 321) appeared to give the best results therefore we used this method.

We then moved to the **preprocessing** of the data.

Our dataset includes 4 numerical ('BMI', 'PhysicalHealth', 'MentalHealth', SleepTime') and

14 categorical features. We have used simple label encoding for binary features with 'Yes', 'No' values and ordinal encoding for features where this was justified, such as age category or general health status of a given patient. Our dataset also includes a categorical feature describing the race of the patient, which was one hot encoded in our dataset.

We used a standard scaler to transform the BMI variable, which was approximately normally distributed, and used a robust scaler for the remaining numerical variables, which were highly skewed.

There were two variables that we did a bit of exploration with: BMI and AgeCategory. We tried to switch the numerical variable BMI to a categorical variable using standard categories (see reference). For AgeCategory, we turned it into a numerical value by taking the middle age of the category. For example if the age group was 55-59, we replaced it by 57. The best results appear when leaving BMI as a numerical. The modification in AgeCategory appeared to have insignificant effects on the models.

5. Evaluation

For this application, we are interested mainly in getting a high recall of the heart disease category while simultaneously minimising the false positives (predicting a healthy patient as unhealthy). The ideal objective is having both a high recall and precision of category 0 and 1. However, it was shown after some tests that both categories won't simultaneously have a recall greater than 75% while still getting a high precision. Thus, to evaluate the performance of the models, we focused on having a high recall for category 1 (predicting correctly patients with heart disease) while maximising the AUC (area under the ROC curve of the true positives - false positives).

5.1 Basic Model Exploration

In the exploratory model selection the aim is to analyse the performance of several models: Naive Bayes, Random Forest, Logistic regression, SVM, and K-Nearest Neighbour. As a result, the maximum AUC score obtained was 0.76 for Logistic Regression and Support Vector Machine classifiers, and a maximum of 79% recall for category 1.

	Category 0 Healthy			Category 1 Unhealthy			
Model	**Prec**	**Recall**	**F1**	**Prec**	**Recall**	**F1**	**AUC**
Complement Naive Bayes	0.97	0.77	0.86	0.23	0.72	0.35	0.75
Random Forest	0.93	0.95	0.94	0.29	0.24	0.26	0.60
Naive Bayes	0.97	0.70	0.82	0.20	0.79	0.32	0.75
Logistic Regression	0.97	0.74	0.84	0.22	0.79	0.34	0.76
SVM	0.97	0.73	0.84	0.22	0.79	0.34	0.76
KNN	0.95	0.80	0.87	0.20	0.54	0.29	0.67

5.2. Ensemble Techniques

The ensemble methods we tested were: XGBoost, AdaBoost, stacking and bagging techniques. The aim of using these techniques was to train weak models and to get a better score in the global prediction. As a result of employing ensemble techniques the result are shown below:

	Category 0 Healthy			Category 1 Unhealthy			
Technique	**Prec**	**Recall**	**F1**	**Prec**	**Recall**	**F1**	**AUC**
XGBoost	0.98	0.69	0.81	0.20	0.82	0.32	0.75
AdaBoost	0.91	0.19	0.32	0.09	0.81	0.15	0.50
Stack: Logit + Naive Bayes	0.98	0.68	0.80	0.20	0.83	0.32	0.76
Bagging	0.98	0.65	0.78	0.18	0.85	0.30	0.75

We trained an XGBoost model, using the negative-to-positive ratio (~10.7) as a class-weighting hyperparameter (scale_pos_weight), to adjust for the significant imbalance in our dataset. This value is used to scale the gradient for the positive class. The best hyperparameters for learning rate, maximum depth, and number of estimators were 0.1, 5, and 10, respectively, with a binary (logistic) objective.

AdaBoost was tuned with a Gaussian Naive Bayes model as the base estimator. The hyperparameter to tune was the number of estimators for the adaptive boosting, this value ranged from 10 to 500. The AUC was used as the objective function. As a result of applying this ensemble technique, the optimizer always tried to maximise one category or the other, but never both at the same time which resulted in poor scores.

For bagging, the best performance was obtained when using a Gaussian Naive Bayes model as the base estimator, and using 50 estimators. For this ensemble technique, the resolver improved the objective pursued, by giving a high recall of the unhealthy category (category 1), and still obtaining a good precision for the healthy category (98%). As a result, the model was able to categorise 6943 (85%) patients with heart disease at the cost of predicting 30955 patients (35%) as unhealthy when they were not.

From all the ensemble techniques performed, the best performance in terms of AUC score was obtained when using the stacking of various models. The combination of first level estimators was based on the performance of each individual model to predict the dataset. From the evaluation of basic models in the model exploration phase, logit and Naive Bayes were among the best models to predict both categories. Indeed, they performed better together than any other combination of models using the stacking technique. Below is the performance of different stacking combinations of models:

		Category 0			Category 1			
First level model Combination	**Final Estimator**	**Prec**	**Recall**	**F1**	**Prec**	**Recall**	**F1**	**AUC**
Random Forest + Naive Bayes + AdaBoost	Naive Bayes	0.93	0.93	0.93	0.25	0.25	0.25	0.59
Logit + Naive Bayes	Naive Bayes	0.98	0.66	0.79	0.19	0.84	0.31	0.76
Logit + KNN + Naive Bayes	Naive Bayes	0.97	0.75	0.84	0.22	0.76	0.34	0.75
Logit + Naive Bayes	Logistic	0.97	0.75	0.84	0.22	0.78	0.35	0.76
Random Forest + Logit + Naive Bayes	Naive Bayes	0.94	0.90	0.92	0.28	0.42	0.33	0.66

For the final estimator, the best performance was obtained when using a Gaussian Naive Bayes, mostly because of its ability to accurately predict the category 0.

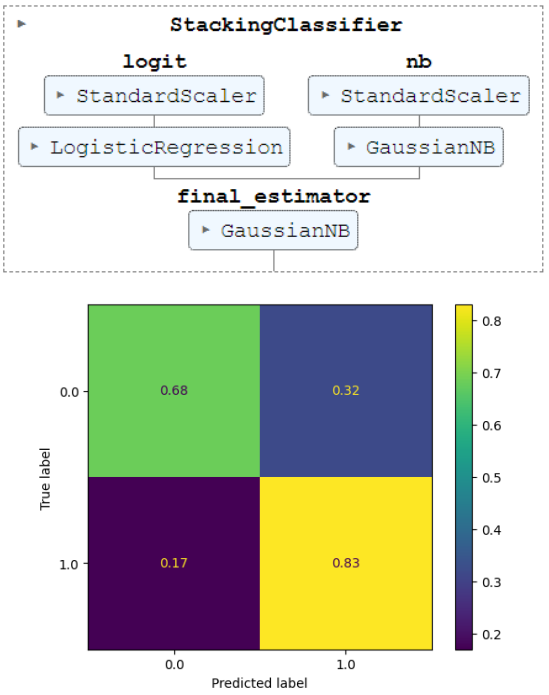

Fig. 2 Stacking Technique configuration and confusion matrix.

As a result, the model correctly predicted 83% of the patients with heart disease at the cost of predicting 32% of the healthy ones as unhealthy.

5.3. Deep Learning

The neural network tested in the present work is based on TensorFlow Keras sequential layers. As the dataset is composed of tabular data, it was decided to create a model with dense layers and dropout layers in between to avoid overfitting. Moreover, the method for tuning the deep learning model was based on performing a random search of hyperparameters for a neural network using Keras Tuner (O'Malley et al., 2019). The hyper parameters tuned were: the number of dense layers, the output shape of each layer, the dropout value of the layers in between, the loss function and the optimizer.

As a result, when using deep learning for solving the binary classification task, the best setup was obtained with the following hyperparameters:

- 5 dense layers, the first two having 352 outputs, one intermediate layer of 192 outputs and two final layers of 352 outputs, in the end a final sigmoid layer was chosen and a decision function for predicting

category 1 when the output is above or equal to 50%.

- To avoid the overfitting during training, three dropout layers were put in between the dense layers with dropout value 0.3.
- The best optimizer and the loss function were binary cross entropy chosen was Nadam which is an Adam algorithm with Nesterov momentum.

With this set-up, the neural network was able to predict 7416 cases out of 8178 as with heart disease while predicting 34497 of 82339 healthy patients as sick when they are healthy. The precision for the 'healthy' category was 98% which represents an AUC of 0.84.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 44)]	0
dense_7 (Dense)	(None, 352)	15840
dropout_6 (Dropout)	(None, 352)	0
dense_8 (Dense)	(None, 352)	124256
dropout_7 (Dropout)	(None, 352)	0
dense_9 (Dense)	(None, 192)	67776
dropout_8 (Dropout)	(None, 192)	0
dense_10 (Dense)	(None, 352)	67936
dropout_9 (Dropout)	(None, 352)	0
dense_11 (Dense)	(None, 352)	124256
dropout_10 (Dropout)	(None, 352)	0

...
Total params: 400,417
Trainable params: 400,417
Non-trainable params: 0

Fig. 3 Best neural network configuration.

	precision	recall	f1-score	support
0	0.98	0.58	0.73	82338
1	0.18	0.91	0.30	8178
accuracy			0.61	90516
macro avg	0.58	0.74	0.51	90516
weighted avg	0.91	0.61	0.69	90516

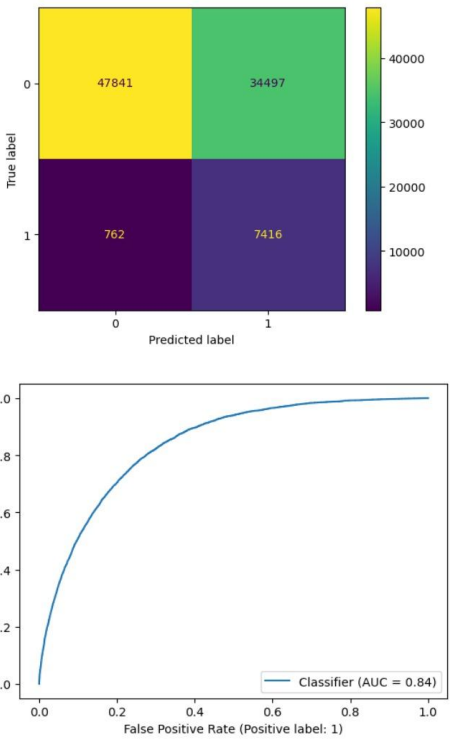

Fig. 4 Neural network result, confusion matrix, and ROC curve.

6. Conclusion

The best performance to correctly predict a patient with heart disease was obtained when training a deep neural network composed of 5 dense layers and 400,417 trainable parameters. The neural network was able to correctly predict 91% of the patients with heart disease from the test set at the cost of predicting 42% of the healthy ones as unhealthy.

The predictions of heart disease obtained during this project can be used as a first screening method and should be used in conjunction with other data that could be obtained from more rigorous exams, specially on the patients predicted as sick before any treatment required. However, this model can give early alerts with a high level of recall for the prediction of heart disease based on 17 features. This way, the patient and medical personnel can trigger actions to confirm and provide treatment if necessary.

While testing various models and techniques to improve the overall performance of the predictions, it was shown that the maximum AUC score

obtained was 0.84. As it is more important to predict a patient with heart disease rather than mis-predicting a healthy one as sick, the best model chosen was still based on getting the AUC score near 0.84 and the highest recall of the unhealthy category.

Despite the high performance of the deep learning model in predicting the unhealthy category, there were still less processor-demanding models such as pure Naive Bayes that obtained a AUC score of 0.76 and still had a recall for the heart disease category of 85%. This can be seen as an attractive result in terms of computer processing efficiency.

References

Abdullah, S. A., & Rajalaxmi, R. R. (2012, April). A Data mining Model for predicting Coronary Heart Disease using Random Forest Classifier. IJCA Proceedings on International Conference in Recent Trends in Computational Methods, Communication and Controls, 22-25.

AL-MILLI, N. (2013, October 10). Back Propagation neural network for prediction of heart disease. Journal of Theoretical and Applied Information Technology.

Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (n.d.). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 321:357.

Dai, W., Theodora S, B., Adams, W. G., Mela, T., Saligrama, V., & Paschalidis, I. C. (2015). Prediction of hospitalization due to heart diseases by supervised learning methods. *International Journal of Medical Informatics*, *84*(3), 189-197. https://doi.org/10.1016/j.ijmedinf.2014.10.002

El-Hasnony, Elzeki, O. M., Alshehri, A., & Salem, H. (2022). Multi-Label Active Learning-Based Machine Learning Model for Heart Disease Prediction. *Sensors (Basel, Switzerland)*, 1184.

Kononenko, I. (n.d.). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, *23*, 89-109.

Mohan, S., Thirumalai, C., & Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access*, *7*, 81542–81554. https://doi.org/10.1109/ACCESS.2019.2923707

O'Malley, T. a. B., Elie and Long, James and Chollet, François and Jin, Haifeng and Invernizzi, & Luca and others. (2019). *KerasTuner*. Keras. Retrieved January 12, 2023, from https://keras.io/keras_tuner/