A Comparison of K Nearest Neighbors (KNN) and Random Forest on predicting ten years risk of coronary heart disease

Leyla Ahmadli

Description and Motivation

Cardiovascular diseases are most common causes of death around the world and one type of it is the coronary heart disease (CHD) [6]. Prognosing CHD in advance can help save the lives of people. Random Forest and KNN classifier will be used to predict having coronary heart disease in 10 years. We will compare and analyze the performance of stated models for classification.

Initial analysis of data set including basic statistics

- The original data set was taken from The Framingham Heart Study Group and a cleaned version is publicly available on Kagle.
- The data set consists of 4133 records, 15 features and a target column.
- There are 7 binary and 8 continuous features in the dataset.
- The binary target column indicates the ten years risk of having CHD. It is 0 for no risk and 1 for having chance of CHD in next ten years. These prognoses were calculated by researchers in The Framingham Heart Study Group, and the data were collected from participants from Framingham, Massachusetts.
- The attributes presented in figure 2 indicate some of the features which have different value ranges. In order to have comparable features, continuous features were normalized to the interval of 0, 1. [1]
- There are high correlations between some attributes as illustrated in figure 3 heatmap. Applying feature selection prior to the training step can increase accuracy of the classification models. Different feature selection methods were used for this step in Random Forest and K Nearest Neighbor classification.
- For K Nearest Neighbor ReliefF method was applied to select features for K Nearest Neighbor classification. ReliefF ranks the predictors according to how well they predict difference of the target classes. [2]
- In figure 1 histogram of TenYearCHD, which is target in our analysis, indicates that there is imbalance in target classes. 85% of the cases have value as 0 and 15% as 1. To address this situation, oversampling minority class was applied to data set to make balanced classes.

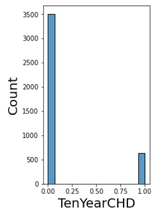


Figure 1. Histogram of target

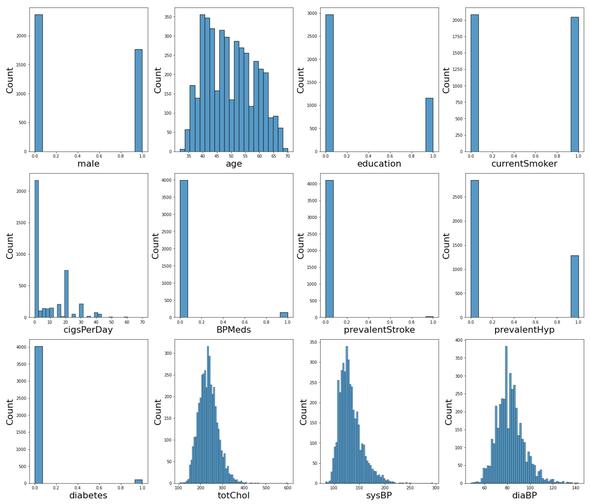


Figure 2. Histograms of predictors

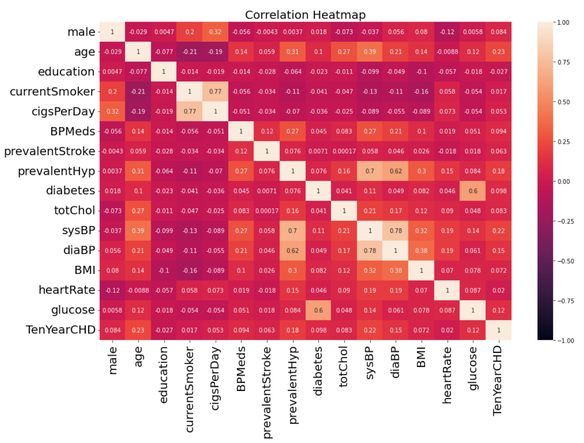


Figure 3.

K Nearest Neighbors

- K Nearest Neighbors is a distance based supervised machine learning model used for classification. Each observation is predicted by calculating distance to locate their K nearest neighbor. The label with highest probability in the located neighbor points is predicted as target class.

Advantages

- It is easily implemented
- KNN is not modeled over distribution of data, it is non-parametric [3]

Disadvantages

- Can overfit the data for lower values of K [3]
- Noisy / irrelevant features can decrease accuracy [3]

Random Forest

- Random Forest is a supervised machine learning model which uses an ensemble learning that constructs several decision trees to sample subsets from entire dataset, and constructs decision trees. In this analysis bagging method was used which is an ensemble of bagged classification trees, which takes bootstrap samples of input data to train each tree.

Advantages

- Less prone to overfitting as it is not using all the input data

Disadvantages

- Time consuming when optimizing hyperparameters and running the model

Hypothesis statement

- We expect that Random Forest will perform better than KNN classification to predict target classes. As it is more complex and uses an ensemble of decision trees for classification.
- We expect that Random Forest will take longer to perform than KNN

Description of the choice of training and evaluation methodology

- Splitting data 70% training and 30% testing sets in order to measure performance of models on unseen testing data set
- Using hyperparameter optimization to decrease error and increase performance
- Calculating accuracy, predict time, F1 score and other measures to analyze overall performance of Random Forest and KNN [4]
- Illustrating feature selection for KNN and predictor importance for Random Forest

Choice of parameters and experimental results

K Nearest Neighbor

- Feature selection with ReliefF to find best predictors for KNN
- Using grid search to find best parameter for 'Distance'

Parameters for K Nearest Neighbors

- Selecting 8 best features from figure 4
- Setting number of nearest neighbors to 3
- Grid search resulted in finding best distance parameter as 'cityblock'

Random Forest

- Using Bayesian Optimization to minimize error of the objective function for allowed range of hyperparameters. These hyperparameter ranges are minimum leaf size of up to 30 and number of predictors to sample up to 15
- Plotting predictor importance parameter

Parameters for Random Forest

- The number of trees in ensemble were selected as 200. Minimum leaf size of 1 was found to be optimal and number of predictors to sample was optimized to 4

	KNN train	RF train	KNN test	RF test
Accuracy	0.903	0.942	0.704	0.786
Precision	0.849	0.868	0.687	0.814
Recall	0.982	0.874	0.760	0.746
F1 score	0.910	0.869	0.721	0.779

Table 1.

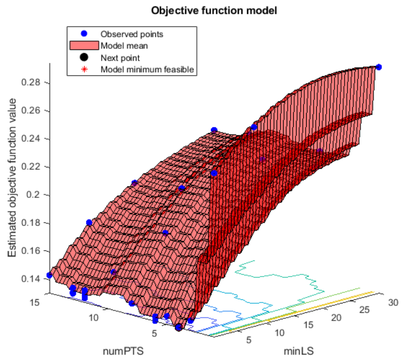
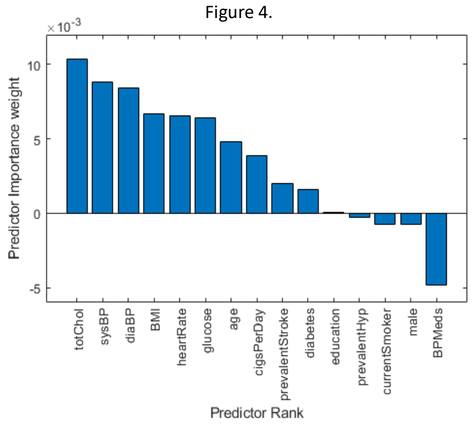


Figure 5.

Figure 4.



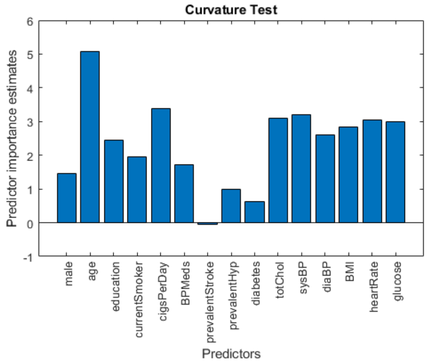


Figure 6.

Figure 7. Confusion Chart for KNN

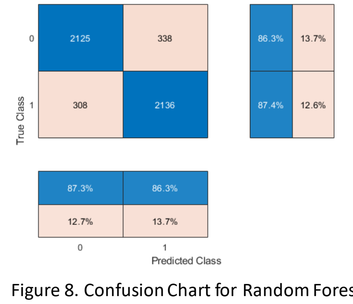


Figure 8. Confusion Chart for Random Forest

Analysis and critical evaluation of results

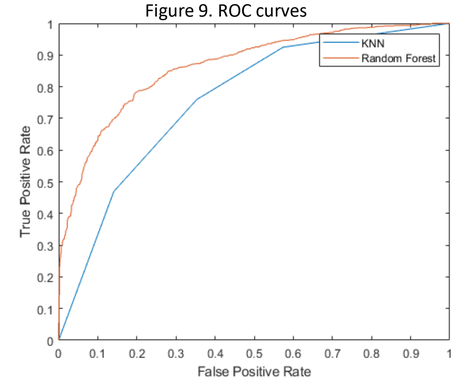
Initial models with imbalanced data gave high accuracy but with very low F1 score. This is understood to be because of models predicting most minority classes as majority. In order to overcome this effect, target classes were balanced by oversampling with synthetic data.

In order to improve the performance of KNN classification we applied feature selection to rank features and select relevant features as seen in figure 4. Additionally, the best predictors for the model were found to be total cholesterol level, blood pressure and BMI. To enhance performance of Random Forest, Hyperparameter Optimizations were applied with Bayesian Optimization [5] to find optimal points for minimum leaf size and number of predictors to sample for each epoch model. In figure 5 we can see Objective function of the model. From the figure we see that objective is low when the number of predictors to sample parameter was greater than 3 and objective decreased as number of minimum leaf size went down. The importance of different predictors for Random Forest can be seen in figure 6; age was found to be the most important factor for Random Forest in the training model.

Timing both models' prediction of test set indicated that on average Random Forest (0.86s) took more time to predict new unseen data than KNN (0.06s).

From figure 7 we see that KNN has a good performance in the training phase, and it is good at predicting positive class rather than negative class. But comparing accuracy and F1 score from Table 1, the performance of model decreases by 20% for testing set. It can mean that there was overfitting for KNN. This can be due to using oversampled data for training the models. From figure 8 we see that Random Forest is more homogenous in the training phase. However, it underperforms when predicting unseen data. Comparing performance of predicting test set of model's accuracy, precision, F1 score for Random Forest was higher than KNN but, recall was high for KNN meaning KNN was good at predicting true positives which in this case is predicting risk of CHD for future ten years. From ROC curves for test set illustrated in figure 9 we see that Random Forest has a higher ratio of true positive to false positives compared to KNN. In conclusion, when looking at the overall performance of the two models, Random Forest was better than KNN.

Figure 9. ROC curves



Lessons learned

- It would have been good to also look at Cross validation error of both models
- Balancing test set was not a good way of choice for predicting future probability of disease because we cannot know how well it will perform in the real world

Future work

- Rather than balancing target class with oversampling in 1:1 ratio it could have been more proper to simultaneously under sample majority class and oversample minority class. When this is to be done, a ratio of under sampling to oversampling should be adjusted such that the probability of selecting from majority and minority classes is comparable. This is to compensate for the initial high disparity between classes.

Reference

1. Pandey, Amit, and Achin Jain. "Comparative analysis of KNN algorithm using various normalization techniques." International Journal of Computer Network and Information Security 11.11 (2017): 36. (http://i.mecs-press.net/ijcnis/ijcnis-v9-n11/IJCNIS-V9-N11-4.pdf)
2. Arora, Nisha, and Pankaj Deep Kaur. "A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment." Applied Soft Computing 86 (2020): 105936. (https://www.sciencedirect.com/science/article/pii/S1568494619307173?casa_token=VldrPwDzVAAAAAA:M7TuAzqTgiKIVBb0XHQl0HqBiiXt7.NKjK6juizhRqlntY0fBoUCcQHfLx7Xvzuk371eBIaFHZg)
3. S. Ray, "A Quick Review of Machine Learning Algorithms," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 35-39, doi: 10.1109/COMITCon.2019.8862451. (https://ieeexplore.ieee.org/abstract/document/8862451)
4. R. Devika, S. V. Avilala and V. Subramaniyaswamy, "Comparative Study of Classifier for Chronic Kidney Disease prediction using Naive Bayes, KNN and Random Forest," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, pp. 679-684, doi: 10.1109/ICCMC.2019.8819654. (https://ieeexplore.ieee.org/document/8819654)
5. N. Quadrianto and Z. Ghahramani, "A Very Simple Safe-Bayesian Random Forest," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 6, pp. 1297-1303, 1 June 2015, doi: 10.1109/TPAMI.2014.2362751. (https://ieeexplore.ieee.org/abstract/document/6920043)
6. https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)