Predicting Heart Disease at Early Stage using Machine Learning

1. Introduction

This capstone project, as part of the data science career track program of Springboard, is aimed to develop a predictive model to detect the risk of heart disease at early stage using health parameters that are typical from routine monitoring. Data wrangling, and exploratory data analysis (EDA) were performed on heart disease dataset from UCI Machine Learning Repository. Supervised machine learning algorithms were trained on the training set with default settings, and there performances were evaluated and compared on the test set. The models with best performances were further tuned using grid search cross validation. Finally, a simple interactive dashboard was built to predict the risk of heart disease based on the best model.

1.1. Problem identification and impact statement

Heart disease is the leading cause of death in the US and worldwide, which produces immense health and economic burdens. In US, more than 600,000 people die from heart disease each year (i.e. 1 in every 4 deaths). The heart disease costs associated with heath care services, medicines, and lost productivity due to death is about $219 billion each year (CDC, 2020). Studies have shown that about 1 in 3 deaths related to heart diseases are preventable if early action is provided (MMWR, 2014). Therefore identifying those at increased risk for heart disease at earliest stage is critical to reduce the mortality associated with heart failure. While genomics, proteomics, and metabolomics have emerged as promising advanced tools to assess risk of disease mechanistically, predictive model based on traditional risk factors (e.g. heath measures typically evaluated in an annual physical) remains an important clinical tool that is rapid, cost-effective, and accurate (Mosley 2020).

1.2. Dataset description

The original database contains 76 attributes, but all published studies refer to using a subset of 14 of them. This project used the Cleveland database downloaded from UCI Machine Learning Repository. A description of each of the 14 attributes is provided in Table 1.

Table 1. Description of dataset

No.	Variable abbreviation	Description
1	age	age in years, continuous
2	sex	sex; 0: female 1: male
3	cp	chest pain type; 1: typical angina 2: atypical angina 3: non-anginal pain 4: asymptomatic
4	trestbps	resting blood pressure, continuous, in mmHg
5	cho	serum cholesterol; continuous
6	fbs	fasting blood sugar; 0: <=120 mg/dL 1: >120 mg/dL
7	restecg	resting electrocardiographic results; 0: normal 1: having ST-T wave abnormal 2: left ventricular hypertrophy
8	thalach	maximum heart rate achieved, continuous
9	exang	exercise induced angina; 0: no 1: yes
10	oldpeak	ST depression induced by exercise relative to rest; continuous
11	slope	the slope of the peak exercise ST segment; 1: upsloping 2: flat 3: downsloping
12	ca	number of major vessels colored by fluoroscopy; integer, 0-3
13	thal	thalassemia (a blood disorder); 3: normal 6: fixed defect 7: reversable defect
14	target	diagnosis of heart disease; 0: no 1: yes

1.3. Methodology

The original data downloaded from UCI Machine Learning Repository was loaded to the python jupyter notebook. The column names were appropriately defined. In addition, we cleaned the raw data by filling missing values, and removing duplicated entries. Further, we examined the data types, counts of unique values, transformed the target values into a binary format, and exported the clean data.

To better understand the data, summary statistics were generated. In addition, the data was visualized via histograms, barplots, boxplots, and a heatmap of correlation matrix.

The cleaned data were appropriately pre-processed for modeling. This included creation of dummy features for categorical variables, splitting the dataset into training and test set, and standardizing the magnitude of numeric features.

Binary classification models were trained on the training set using default settings. The model performance were evaluated using the test set. Models with high performance were selected for hyperparameter tuning via grid search. The 'recall' score was used for optimization, in order to minimize false negatives during prediction of heart disease. The model with highest recall score as well as high accuracy and roc-auc score was selected as the final model to make future predictions.

2. Data wrangling

Among all data, we found two variables that have missing values as represented by question markers. The missing values were replaced with null and filled using the mode of each variable. No duplicated entries were found. The response variable 'target' was converted into a binary format: 0 for absence of heart disease, and 1 for presence of heart disease. The cleaned data contains 303 rows and 14 columns, which was exported for subsequent analyses. The data cleaning process is detailed in the jupyter notebook (heart_disease_ML.ipynb).

3. Exploratory data analysis

The statistics including mean, standard deviation, minimun values, maximum values, and percentiles for each variable were summarized in Table 2.

Table 2. Summary statistics

	count	mean	std	min	25%	50%	75%	max
age	303	54.44	9.04	29	48	56	61	77
sex	303	0.68	0.47	0	0	1	1	1
cp	303	3.16	0.96	1	3	3	4	4
trestbps	303	131.69	17.60	94	120	130	140	200
chol	303	246.69	51.78	126	211	241	275	564
fbs	303	0.15	0.36	0	0	0	0	1
restecg	303	0.99	0.99	0	0	1	2	2
thalach	303	149.61	22.88	71	133.5	153	166	202
exang	303	0.33	0.47	0	0	0	1	1
oldpeak	303	1.04	1.16	0	0	0.8	1.6	6.2
slope	303	1.60	0.62	1	1	2	2	3
ca	303	0.66	0.93	0	0	0	1	3
thal	303	4.72	1.94	3	3	3	7	7
target	303	0.46	0.50	0	0	0	1	1

The distributions of numerical variables were visualized in figure 1. Unique values and their counts for each categorical variables were visualized in figure 2.

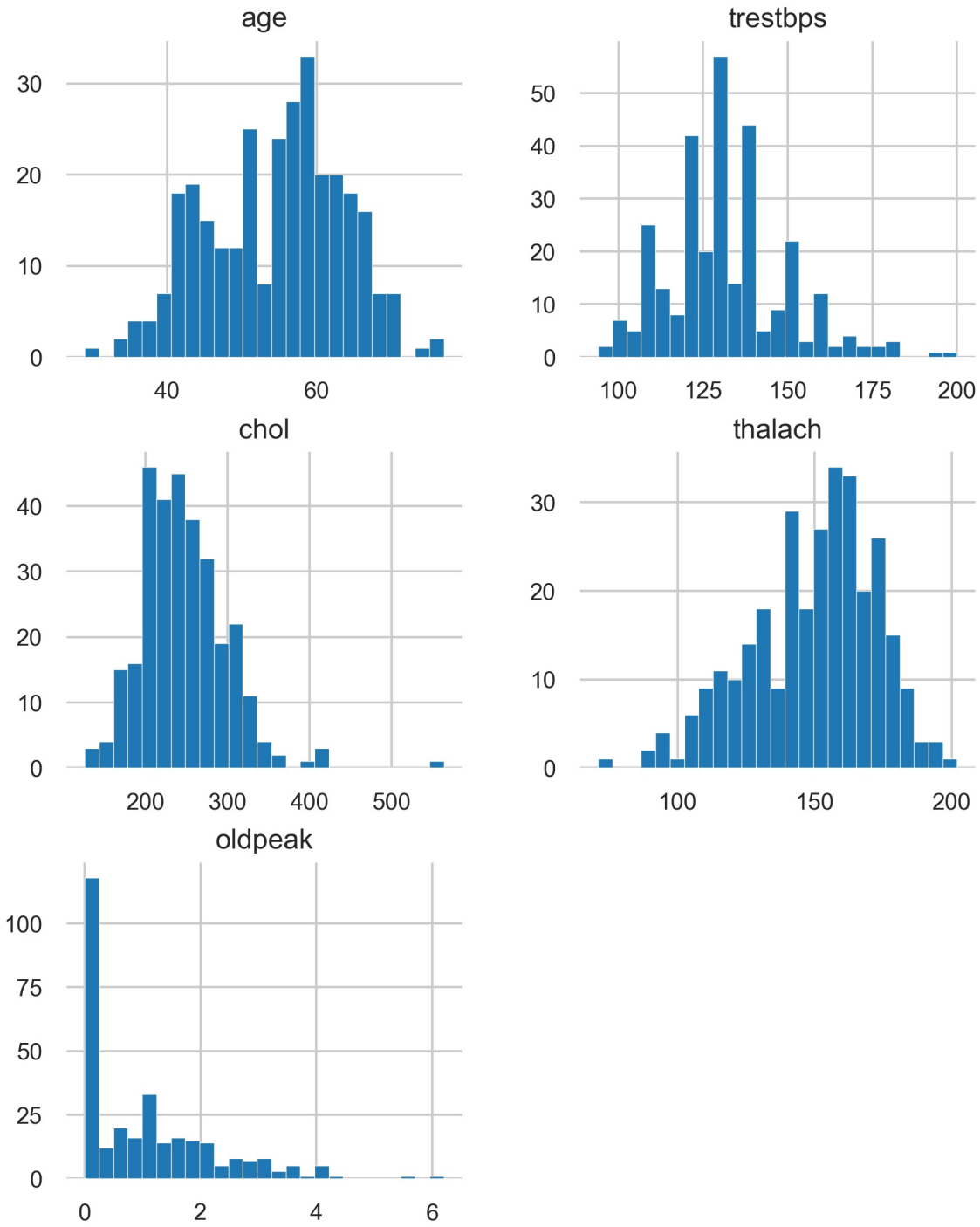

Figure 1. Histograms showing the distribution of numerical variables.

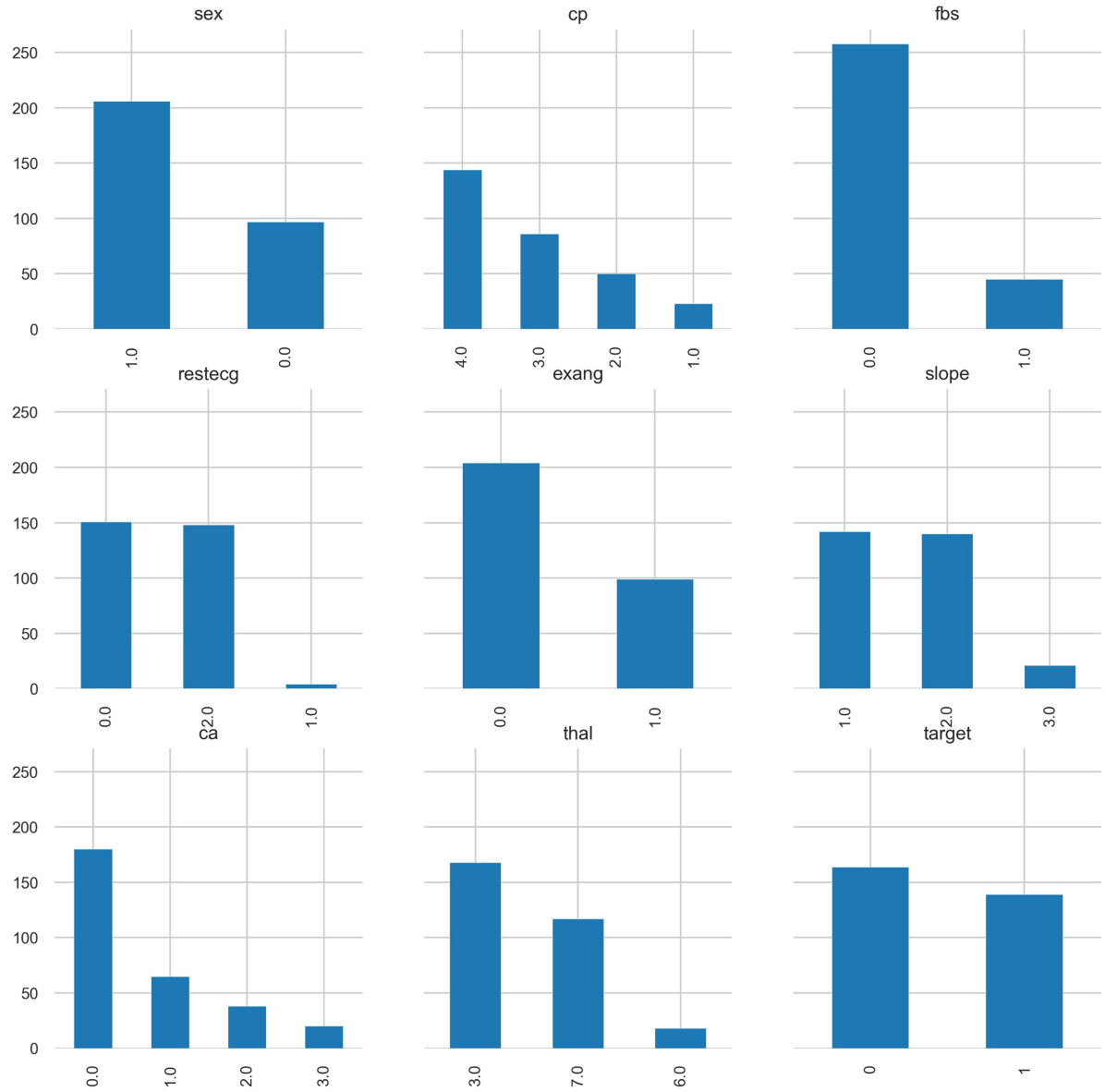

Figure 2. Barplots of categorical variables.

A few extreme values for variables (oldpeak, thalach, chol, and trestbps) were detected as shown in circles (i.e. outside of 1.5*IQR) in the boxplots (Fig. 3).

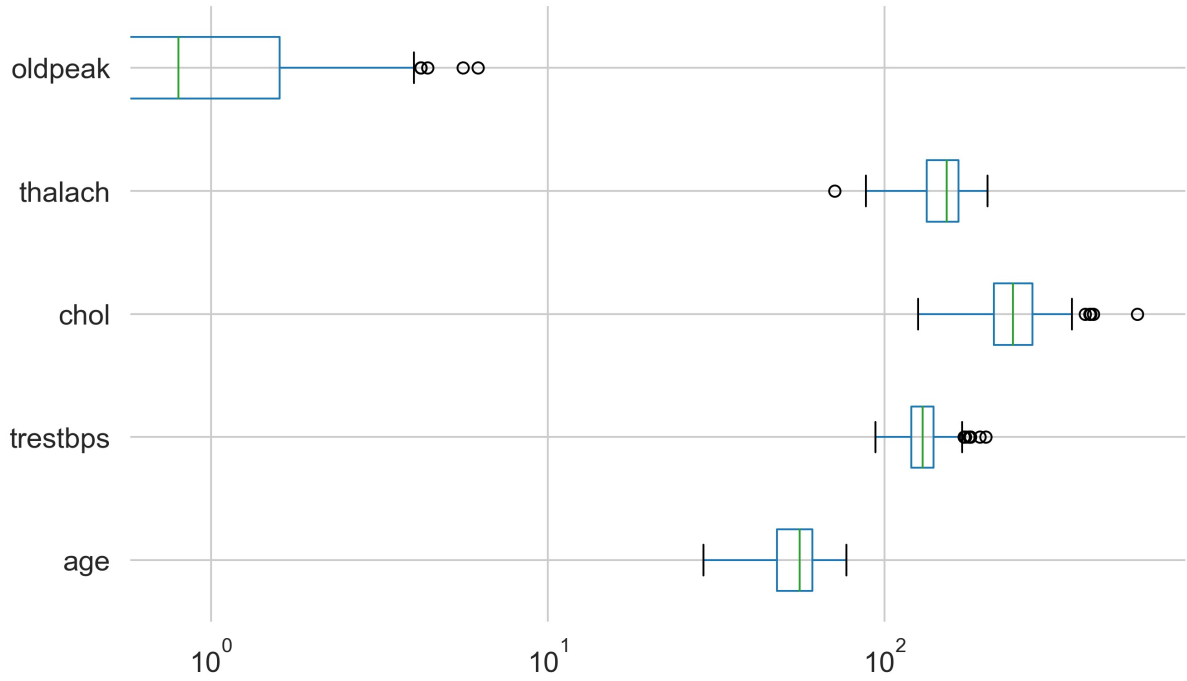

Figure 3. Boxplots of numerical variables. Circles represent extreme values/outliers of each variable.

No significant correlation coefficient was found among variables (Fig. 4). This is probably because the original dataset has been processed to only contain 14 variables as mentioned in the Introduction.

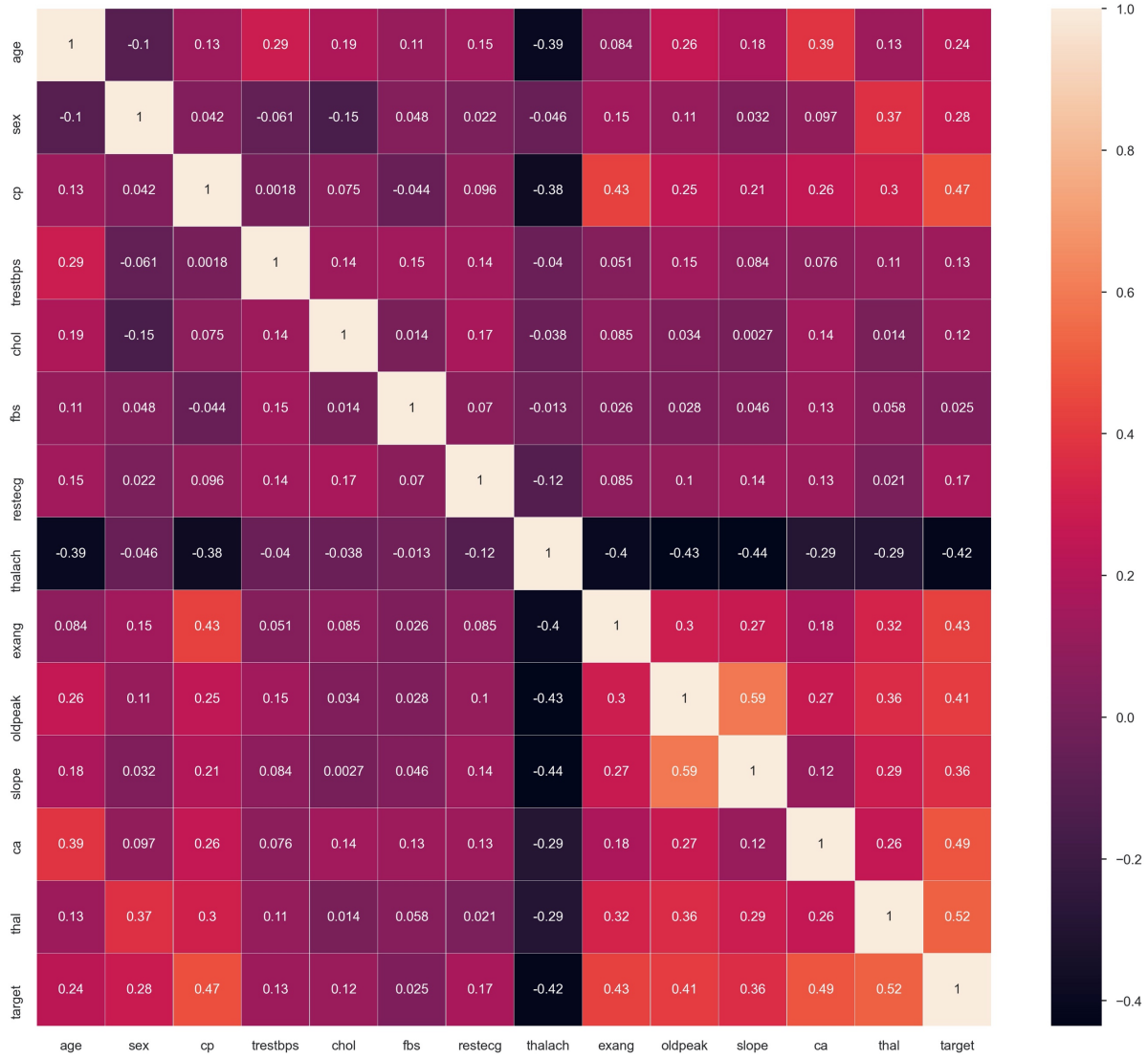

Figure 4. Heatmap of correlation coefficients between variables.

4. Pre-processing and training data preparation

Dummy features were created for categorical variables (sex, cp, restecg, exang, and slope). As the number of major vessels (ca) is ordinal, we did not created dummy features for ca. This process increased the number of features from 13 to 19. Training and test set were created for model input. The ratio of classes in the original dataset were maintained by specifying stratification during data split. Further, the magnitude of numerical features in the training set were standardized to have 0 mean and 1 variance. The same scaler was applied to the test set. More details were provided in the jupyter notebook (heart_disease_ML.ipynb).

5. Modeling

5.1. An overview of binary classifiers

Table 3. A brief summary of advantage and disadvantage of binary classifiers.

Binary Classifier		Advantages	Disadvantages
Naïve Bayes		Simple, fast, low computation cost, and accurate	Cannot learn interactions between features
Logistic Regression		Lots of ways to regularize the model (e.g. lasso, ridge), and don't have to worry as much about features being correlated, like in Naive Bayes; have a nice probabilistic interpretation; feature scaling is not a requirement	Poor performance on non-linear data (e.g. images)
k-Nearest Neighbors		No assumptions about data; simple and intuitive, relatively high accuracy; constantly evolving model	Curse of dimensionality; feature scaling is an absolute must; does not perform well on imbalanced data; sensitive to outliers; slow for large dataset
Support Vector Machine		Good performance over high-dimension data (e.g. images), and is not sensitive to outliers	Poor performance with overlapping classes, and is sensitive to the type of kernel used; hyperparameter tuning is important
Decision Tree		Feature scaling is not needed; Easy to explain and visualize	Prone to overfitting
Ensemble - bagging	Random Forest	Easy to interpret and explain; can handle feature interactions, non-parametric; fast and scalable	Don't support online learning
Ensemble - boosting	AdaBoost	Low generalization error, easy to implement, and works with many classifiers	Sensitive to outliers
	Gradient boosting	High accuracy and flexibility	Sensitive to outliers and computationally expensive
	XGBoost	Feature scaling is not needed; computational efficiency and often better model performance	Difficult to interpret and visualize; hard to tune (a lot hyperparameters)
	LightGBM	high speed, high accuracy, can use categorical features as input directly	Prone to overfitting

5.2. Models with default settings

The above mentioned 10 models (i.e. gaussian naive bayes classifier, logistic regression, k-nearest neighbors, support vector machine, decision tree, random forest, adaptive boost, gradient boosting, XGBoost, LightGBM) were trained using the training set, with default settings to the training set. The model performance was evaluated on the test set, via metrics including precision, recall, accuracy, matthews correlation coefficient, and area under ROC. More details are provided in the jupyter notebook (heart_disease_ML.ipynb).

5.3. Model evaluation

A comparison of model performance is provided (Table 4, and Fig.5). From figure 5, we can clearly see that most of the models have roc-auc score above 0.8. Models that have accuracy scores above 0.8 only included logistic regression, random forest, and XGBoost, among which, only logistic regression and random forest have recall scores above 0.8.

Table 4. A comparison of model performance

model	precision	recall	accuracy	mcc	roc_auc
GaussianNaiveBayes	0.725	0.829	0.776	0.559	0.843
LogisticRegression	0.811	0.857	0.842	0.685	0.879
kNN	0.737	0.8	0.776	0.554	0.831
SVM	0.788	0.743	0.789	0.575	0.858
DecisionTree	0.757	0.8	0.789	0.579	0.79
RandomForest	0.848	0.8	0.842	0.682	0.892
AdaBoost	0.765	0.743	0.776	0.549	0.862
GradientBoosting	0.794	0.771	0.803	0.602	0.872
XGBoost	0.818	0.771	0.816	0.629	0.877
LightGBM	0.765	0.743	0.776	0.549	0.845

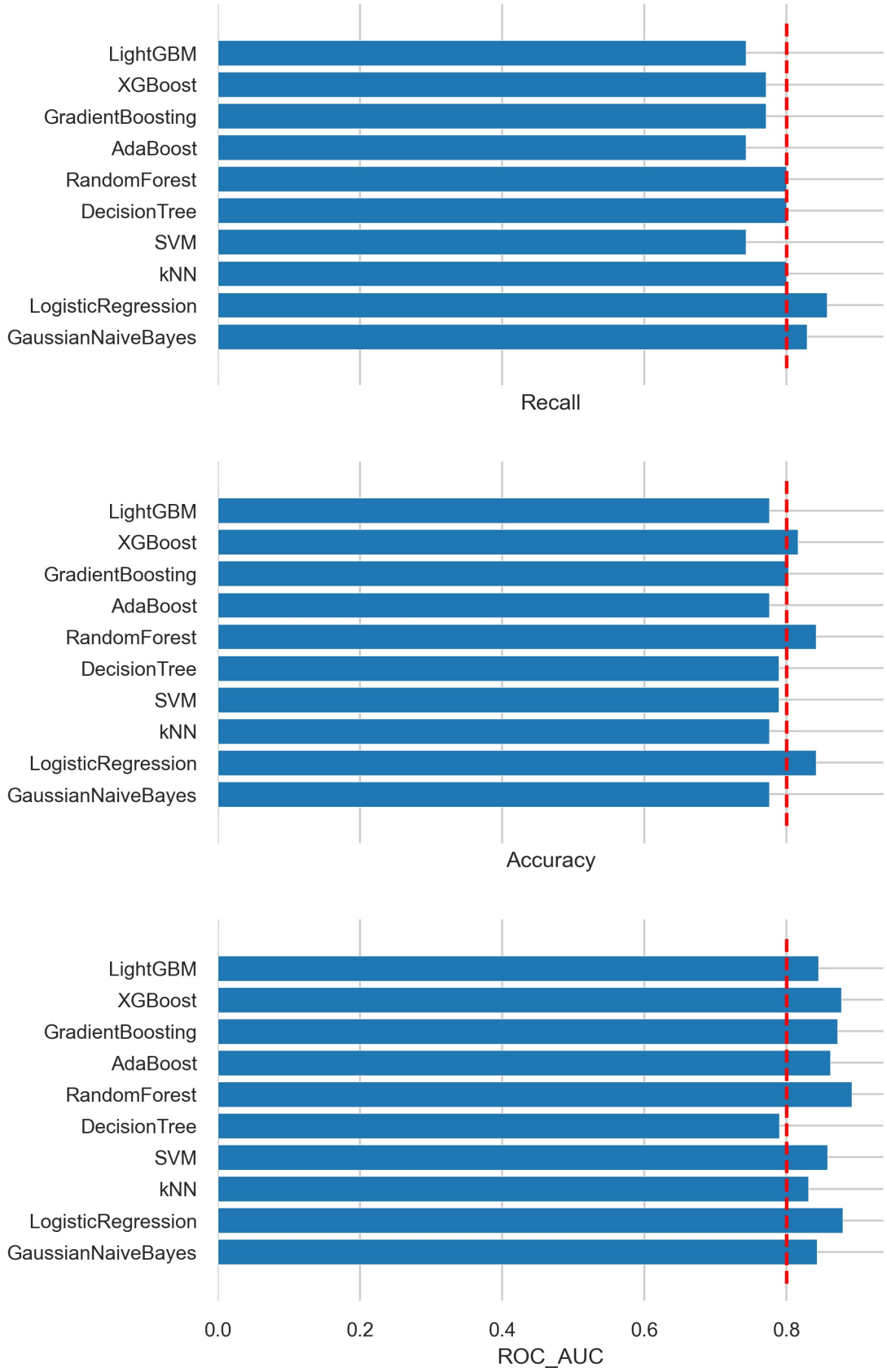

Figure 5. Barplots comparing metrics of model performance.

5.4. Model optimization

Hyperparameter tuning via grid search cross validation was applied to logistic regression, and random forest. In order to minimize false negatives in predictions, we set recall score as the score to maximize during tuning. More details are provided in the jupyter notebook (heart_disease_ML.ipynb).

A comparison of model performance with and without hyperparameter tuning is provided in the table below.

Table 5. Model performance with and without hyperparameter tuning.

model	precision	recall	accuracy	mcc	roc_auc
LogisticRegression	0.811	0.857	0.842	0.685	0.879
LogisticRegression_tuned	0.811	0.857	0.842	0.685	0.882
RandomForest	0.848	0.8	0.842	0.682	0.892
RandomForest_tuned	0.853	0.829	0.855	0.708	0.886

5.5. Final model

Hyperparameter grid search further increased the performance of random forest and logistic regression, although there was only slight increases in the performance of logistic regression (roc_auc score increased from 0.879 to 0.882). After optimization, random forest gives the highest accuracy (i.e. 0.855), and logistic regression gives the highest recall (i.e. 0.857). Logistic regression with tuned hyperparameters (C=1.0, penalty='l1', solver='liblinear') was selected as the final model for prediction of heart disease risks. This is because logistic regression, although gives slightly lower accuracy score (i.e. 0.842) compared to random forest, it gives the highest recall score (i.e. 0.857). A higher recall score means lower false negatives, which is prefered in the prediction of heart disease. In addition, compared to random forest, logistic regression is substantially faster, and it is easier to incorporate more training data once they become available.

The ROC curve of the final model - logistic regression was provided in Fig.6. In addition, relative importance of features based on logistic regression was compared in Fig.7.

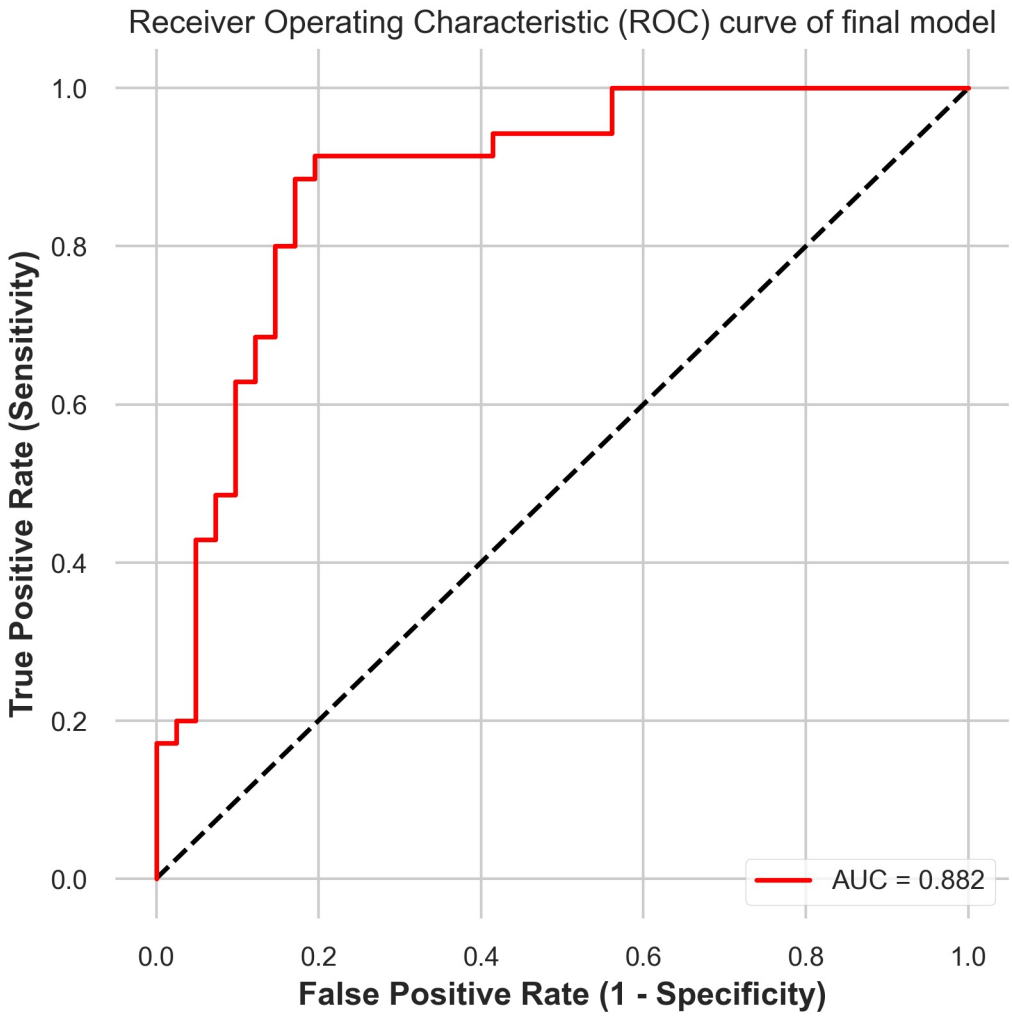

Figure 6. Receiver operating characteristic curve of logistic regression (C=1.0, penalty='l1', solver='liblinear').

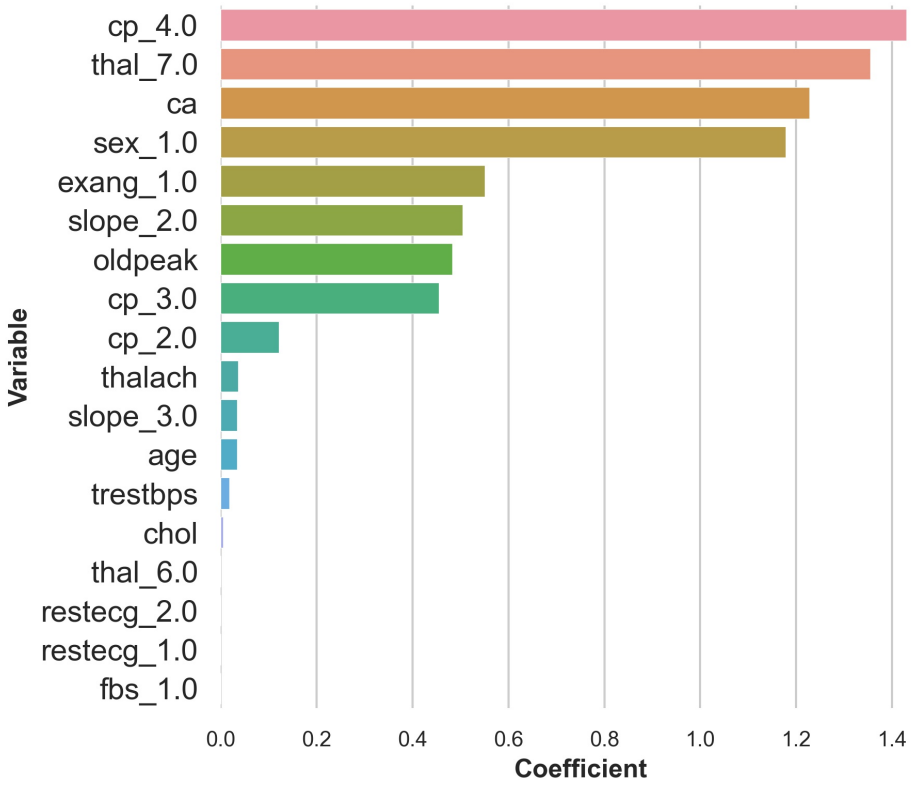

Figure 7. Relevant importance of features.

6. Conclusions

Heart disease is the leading cause of death in the US and worldwide, which produces immense health and economic burdens. Identifying those at increased risk for heart disease at earliest stage is critical to reduce the mortality associated with heart failure. While advanced 'omics' technologies have emerged as promising ways of understanding risk of disease mechanistically, predictive model based on traditional risk factors remains an important clinical tool that is rapid, cost-effective, and can be as accurate as molecular methods.

A comparison of 10 binary classification algorithms suggested that logistic regression and random forest were among the models with top performance. Subsequent hyperparameter tuning via grid search further increased the model performance. Although random forest with grid search hyperparameter tuning gives the highest accuracy (i.e. 0.855), logistic regression (with tuned hyperparameters: C=1.0, penalty='l1', solver='liblinear') was selected as the final model for prediction of heart disease risks. This is because logistic regression, although gives slightly lower accuracy score (i.e. 0.842) compared to random forest with grid search, it gives the highest recall score (i.e. 0.857). A higher recall score means lower false negatives, which is important in terms of predicting heart disease. In addition, compared to random forest, logistic regression is substantially faster, and it is easier to incorporate more training data once they become available.

In future, it would be worthwhile to do additional feature engineering provided with more domain knowledge and data. Developing a user-friendly interactive dashboard tool is equally important.

Acknowledgements

This work is not possible without the database provided by the UCI Machine Learning repository. I would like to thank Dipanjan Sarkar, Kenneth Gil-Pasquel, and the Springboard team who provided inputs and guidance during this project.

References

CDC, 2020. Heart Disease Statistics and Maps. URL: https://www.cdc.gov/heartdisease/facts.htm (assessed Aug. 19, 2020)

MMWR, 2014. Potentially Preventable Deaths from the Five Leading Causes of Death — United States, 2008–2010. URL: https://www.cdc.gov/mmwr/preview/mmwrhtml/mm6317a1.htm (assessed Aug. 19, 2020)

Mosley, J., D. Cupta, J. Tan. Predictive Accuracy of a Polygenic Risk Score Compared With a Clinical Risk Score for Incident Coronary Heart Disease. JAMA. 2020;323(7):627-635. doi:10.1001/jama.2019.21782