IBM ML course - Classification project report

- Predicting heart disease with machine learning

Major objective of the analysis

For this exercise, we will be using the heart disease dataset from UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/45/heart+disease). The major objective is to predict the presence of heart disease using clinical parameters. The models could help doctors select patients for further assessment and follow-up.

We will train four classifiers (logistic regression, support vecter machine, random forest, and XGBoost) for their difference in complexity, training speed and explainability. with the same train and test splits and compare their prediction performance. The results will be used for recommending a final model for this dataset. Note that this dataset (303 clinical cases; 164 cases didn't have heart disease while 139 cases have) is relatively small, therefore the models may not generalize well to other datasets, but they can be used as starting points.

The dataset

The heart disease repository includes four databases: Cleveland, Hungary, Switzerland, and the VA Long Beach. According to data information, "This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to date." Therefore, we will be using the Cleveland database with 14 features, "simply attempting to distinguish presence (values 1, 2, 3, 4) from absence (value 0)" as other researchers did. Specifically, we will be using the processed Cleveland data which is "in good shape and is usable (for the 14 attributes situation)" because the original one was messed up due to computer failure (see the "WARNING" in downloaded folder).

The Cleveland dataset includes 303 patients with the following clinical information (Table 1 for a quick look):

1. sex - (1 = male; 0 = female)
2. age - age in years
3. cp - chest pain type (1 = typical angina, 2 = atypical angina, 3 = non-anginal pain, 4 = asymptomatic)
4. trestbps - resting blood pressure in mmHg on admission to the hospital

5. chol - serum cholesterol in mg/dl
6. fbs - fasting blood sugar > 120 mg/dl (1 = true, 0 = false)
7. restecg - resting electrocardiographic results (0 = normal, 1 = having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)
8. thalach - maximum heart rate achieved
9. exang - exercise induced angina (1 = yes; 0 = no)
10. oldpeak - ST depression induced by exercise relative to rest
11. slope - the slope of the peak exercise ST segment (1 = upsloping, 2 = flat, 3 = downsloping)
12. ca - number of major vessels (0-3) colored by flourosopy
13. thal – thalassemia (3 = normal, 6 = fixed defect, 7 = reversable defect)
14. num - diagnosis of heart disease (0 = no, 1-4 = yes)

Table 1. The column names and the first five rows of the dataset

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

Data exploration, cleaning and feature engineering

Below we will look at the data in details, dealing with missing data, and look at the correlation between features and target.

Data types. Original data has categorical and numeric data, but because this set is a processed dataset, all the features are now integer (11) or float type (3), as shown below from data information output of Jupyter Notebook:

```
Data columns (total 14 columns):
#     Column    Non-Null Count  Dtype
---   --------  ----------------- ------
 0   age        303 non-null     int64
 1   sex        303 non-null     int64
```

```
 2   cp         303 non-null    int64
 3   trestbps   303non-null   int64
 4   chol       303non-null   int64
 5   fbs        303non-null   int64
 6   restecg    303non-null   int64
 7   thalach    303non-null   int64
 8   exang      303non-null   int64
 9   oldpeak    303non-null   float64
 10  slope      303non-null   int64
 11  ca         299non-null   float64
 12  thal       301non-null   float64
 13  num        303non-null   int64
dtypes: float64(3), int64(11)
```

Basic description of the data (Table 2). As typically seen, clinical test results (such as trestbps, chol, thalach, and oldpeak) can have wide ranges. Feature values are on various scales.

Table 2. Basic description of the features

	age	sex	cp	restbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
count	303	303	303	303	303	303	303	303	303	303	303	299	301	303
mean	54.44	0.68	3.16	131.69	246.69	0.15	0.99	149.61	0.33	1.04	1.60	0.67	4.73	0.94
std	9.04	0.47	0.96	17.60	51.78	0.36	0.99	22.88	0.47	1.16	0.62	0.94	1.94	1.23
min	29.00	0.00	1.00	94.00	126.00	0.00	0.00	71.00	0.00	0.00	1.00	0.00	3.00	0.00
25%	48.00	0.00	3.00	120.00	211.00	0.00	0.00	133.50	0.00	0.00	1.00	0.00	3.00	0.00
50%	56.00	1.00	3.00	130.00	241.00	0.00	1.00	153.00	0.00	0.80	2.00	0.00	3.00	0.00
75%	61.00	1.00	4.00	140.00	275.00	0.00	2.00	166.00	1.00	1.60	2.00	1.00	7.00	2.00
max	77.00	1.00	4.00	200.00	564.00	1.00	2.00	202.00	1.00	6.20	3.00	3.00	7.00	4.00

Missing data and treatment. There are missing values: 2 in thal (Table 3), and 4 in ca (Table 4). The dataset is relatively small as are the missing values, so the missing values were filled using pandas.fillna() function to keep the samples (Table 5 and Table 6).

Table 3. Missing data in thal feature

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	um
87	53	0	3	128	216	0	2	115	0	0.0	1	0.0	NaN	0
266	52	1	4	128	204	1	0	156	1	1.0	2	0.0	NaN	2

Table 4. Missing data in ca feature

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
166	52	1	3	138	223	0	0	169	0	0.0	1	NaN	3.0	0
192	43	1	4	132	247	1	2	143	1	0.1	2	NaN	7.0	1
287	58	1	2	125	220	0	0	144	0	0.4	2	NaN	7.0	0
302	38	1	3	138	175	0	0	173	0	0.0	1	NaN	3.0	0

Table 5. Missing data in thal feature (after filled)

	ge	ex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
87	53	0	3	128	216	0	2	115	0	0.0	1	0.0	3.0	0
266	52	1	4	128	204	1	0	156	1	1.0	2	0.0	6.0	2

Table 6. Missing data in ca feature (after filling)

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
166	52	1	3	138	223	0	0	169	0	0.0	1	0.0	3.0	0
192	43	1	4	132	247	1	2	143	1	0.1	2	3.0	7.0	1
287	58	1	2	125	220	0	0	144	0	0.4	2	2.0	7.0	0
302	38	1	3	138	175	0	0	173	0	0.0	1	1.0	3.0	0

“Outliers” and data skews. “Outliers” are seen in some features particularly chol and oldpeak (Figure 1), but these are most likely biological, therefore being left as are. Some skews can be seen in chol (1.1, right skew; Figure 2), age (-0.21, left skew), and trestbps (0.7, right skew). Again, these could be biological. We will normalize the data before splitting.

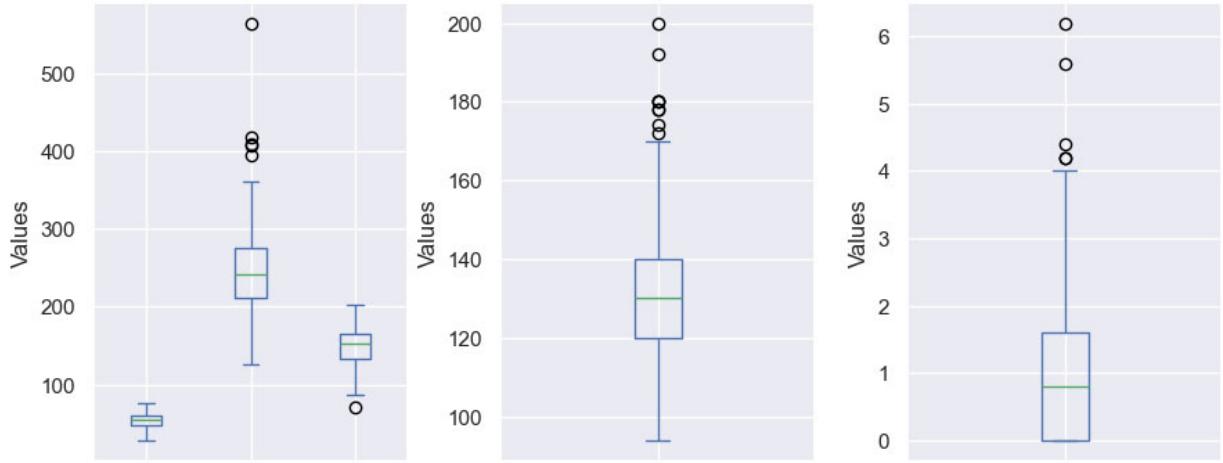

Figure 1. Box plot of age, trestbps, chol, thalach, and oldpeak

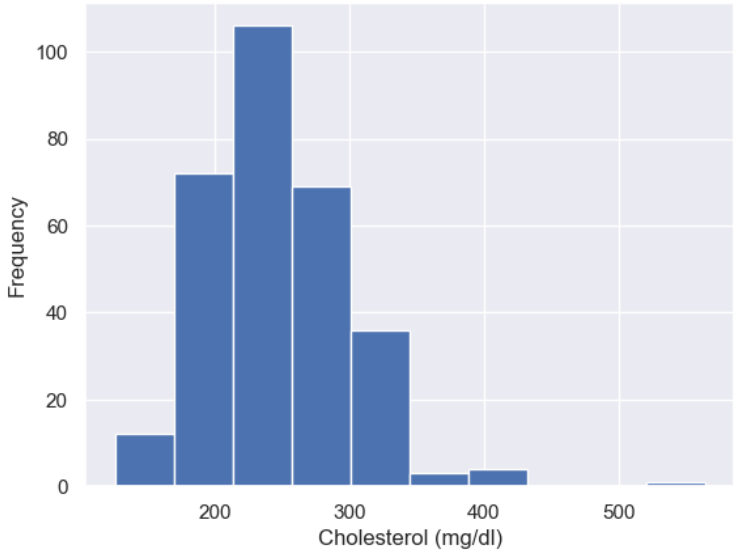

Figure 2. Distribution of cholesterol values

Feature correlations (Figure 3). From the heatmap, it appears that thal, ca, exang, oldpeak, and cp have > 0.4 positive, while thalach stands out as having – 0.4, correlation with disease. Some correlations can be found between other features (slope with oldpeak for example).

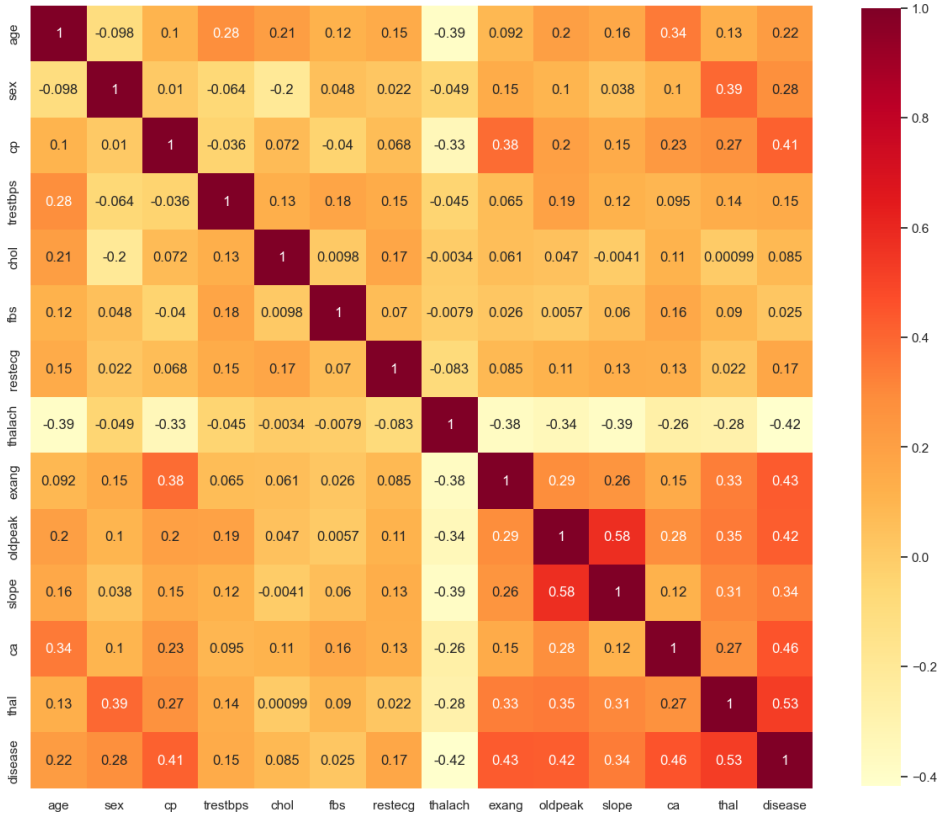

Figure 3. Heatmap of feature correlation.

Feature engineering. The features can be put into three categories: continuous (age, trestbps, chol, thalach, ca, and oldpeak); binary (sex, fbs, and exang); and ordinal (restecg, cp, and slope). They are already numeric. Thal is unique in values (3 = normal, 6 = fixed defect, 7 = reversable defect) with unclear meanings for their scales. For simplicity, we changed the values (3 to 0, 6 and 7 to 1), turning it into a binary feature. The num (target) was ordinal. We changed the name to “disease” for clarity with all values other than 0 changed to 1 (indicating presence of the disease), also turning it into binary as in published studies. Finally, data were scaled with StandardScaler.

All features will be used for modeling as previous studies did.

Class balance. The cases with or without heart disease are 164 and 139, respectively, therefore the target classes are relatively balanced; stratification will be used when splitting the data.

Model training, evaluation metrics and comparison

The data was split into X (features) and y (target), and then X was scaled with StandardScaler. X and y were then split into training and test sets stratification (test size = 0.25). These training and test sets were used for training and prediction of four classifiers (linear regression, support vector machine (SVM), random forest, and XGBoost classifiers). For each classifier, sklearn GridSearchCV cross-validation was used to optimize hyperparameters. The four models with their best hyperparameters (as included in GridSearchCV) are summarized below (data split: test size = 0.25, stratification) :

Logistic regression:
‘penalty’: 'l2', ‘solver’: 'liblinear';

Support vector machine:
'C': 0.1, 'kernel': 'sigmoid';

Random forest:
'max_depth': 5, 'max_features': 0.3, 'max_samples': 0.7, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 63;

XGBoost:
'learning_rate': 0.6, 'max_depth': 1, 'n_estimators': 27.

The scoring results were collected in Table 7 and confusion matrix in Table 8.\

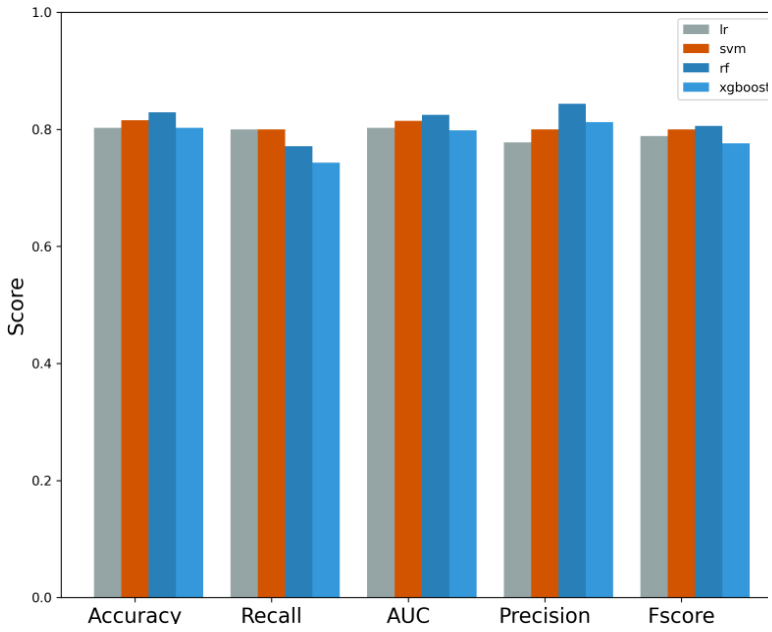

Figure 4. Evaluation metrics of the four classifiers

Table 7. Evaluation metrics of the four models

classifier	accuracy	recall	AUC	precision	f1 score
Linear regression	0.80	0.80	0.80	0.78	0.79
SVM	0.82	0.81	0.80	0.80	0.80
random forest	0.80	0.80	0.74	0.81	0.78
XGBoost	0.82	0.81	0.74	0.84	0.79

Table 8. Confusion matrices of the four models (+: positive; -, negative)

classifier	linear regression		SVM		random forest		XGBoost	
predicted	-	+	-	+	-	+	-	+
true -	33	8	34	7	35	6	36	5
true +	7	28	7	28	9	26	9	26

Performance. Table 7 and Figure 4 show the evaluation metrics of the models. While overall, the performances of the four models are largely similar on this dataset, examined closely, random forest classifier scored the highest in accuracy, AUC, precision and f1 score, but as was XGBoost classifiers, it suffered in recall (XGBoost being the lowest) than the other two classifiers. SVM did a slightly better than logistic regression in all scores except recall, where they were equal.

Consistently, the confusion matrices show that, for this dataset, logistic regression and SVM classifier produced very close results; that while random forest and XGBoost classifiers have lower false positives, they tend to have higher false negatives in comparison to logistic regression and SVM classifiers.

Feature importance. The results of the training set were presented in Figure 5 to Figure 8. As revealed by sklearn feature permutaion_importance for the training set, the first four most importance features to each classifier are: ca, thal, sex, and cp for logistic regression; thal, ca, cp, and thalach for SVM; thal, cp, exang, and ca for random forest; and oldpeak, ca, cp, and thal for XGBoost. Therefore, while different models look at the feature differently, they agree the importance of thal (thalassemia), ca (number of major

vessels colored by flouroscopy), and cp (chest pain type) for training the models. The three features were also important for prediction of the models as assessed with permutation_importance with the test set.

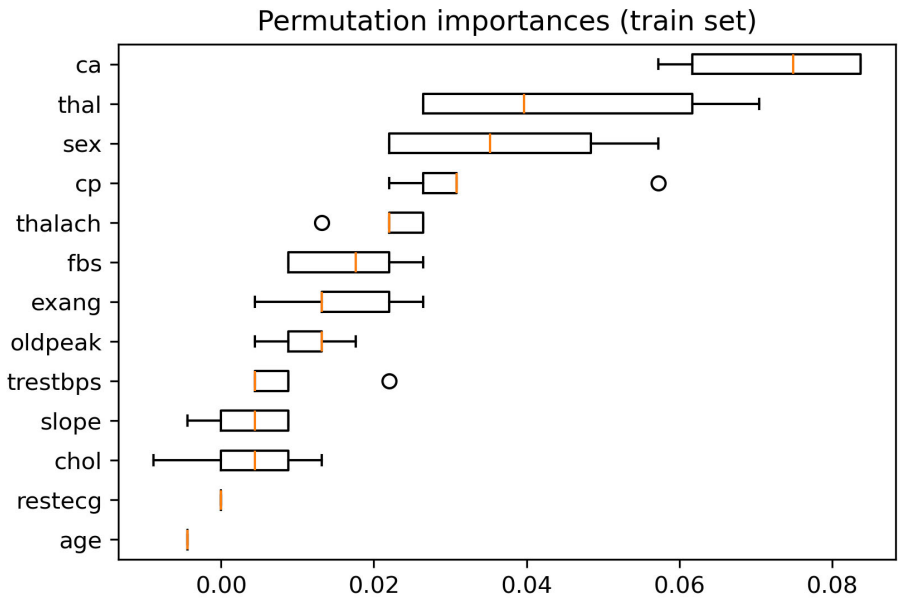

Figure 5. Feature importance for training of logistic regression

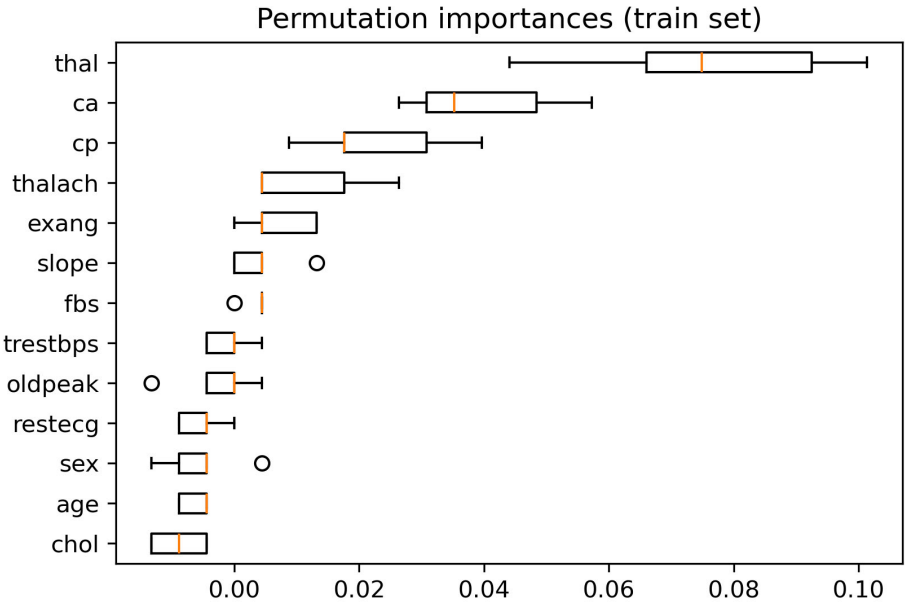

Figure 6. Feature importance for training of SVM classifier (SV

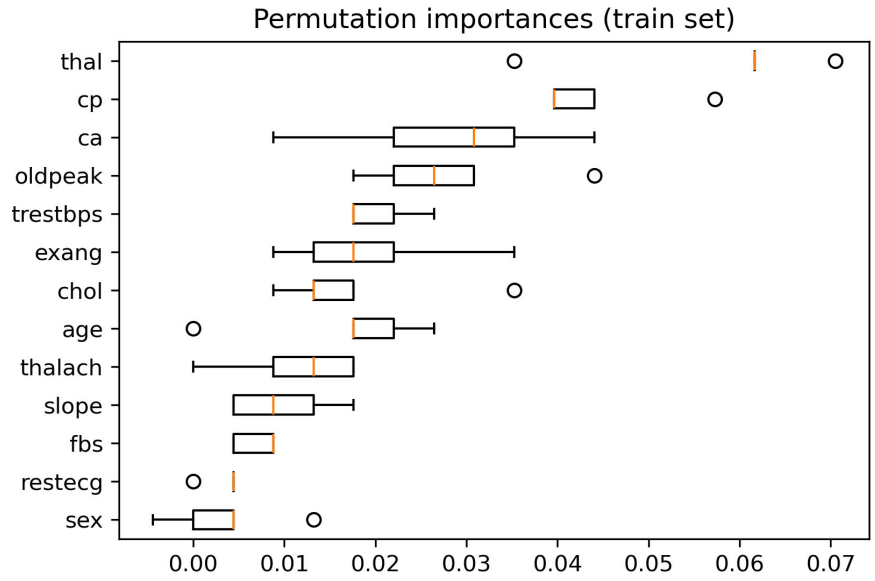

Figure 7. Feature importance of random forest classifier

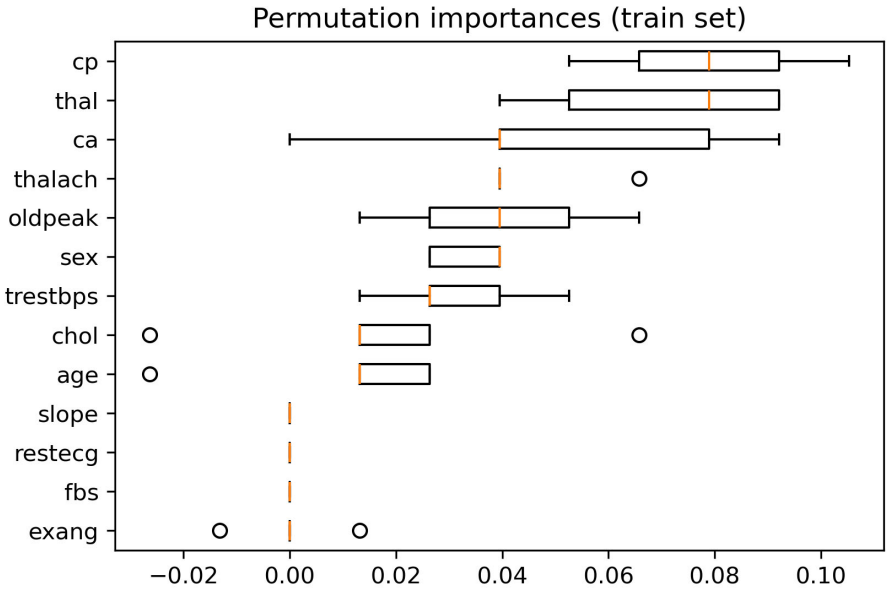

Figure 8. Feature importance of XGBoost classifier

Final model recommendation

Given the above model performance results, the SVM classifier (SVC) is recommended as the final model for its short training time and relatively good explainability. While

logistic regression has better explainability than SVM, its overall performance was slightly worse than SVM. Random forest and XGBoost are not recommended because of their low recalls; we would prefer higher recall than higher precision for this type of disease prediction. We would choose potential patients for further medical evaluations rather than missing cases.

Summary of key findings and insights

The four classifiers trained have mostly similar prediction performance with ~ 0.8 prediction accuracy; random forest and XGBoost appeared to have slightly high precision (0.81, 0.84, respectively) but lower recall than logistic regression and SVM (0.78, 0.8, respectively). Overall SVM performed slightly better than logistic regression, therefore is recommended as the final model. The two esemble models were not recommended because they tend to prodict more false negatives, which is not desirable clinically. Three features, thal (thalassemia), ca (number of major vessels colored by flouroscopy), and cp (chest pain type), were identified as the most importance feature in common for these models, which are clinically relevant.

Further data analysis and model improvement

The models could be improved by further feature engineering (add polynomial features for logistic regression for example, which was suggested by the performance of SVM), fine tuning of the parameters, and removing those features deemed unimportant by the models. Because the original dataset has 76 attributes, it is possible to include some of them guided by clinical evidence. Further, the dataset deposit includes three sets of data beside the one used here, which are worthy of exploration. Ultimately, as heart diseases are complex and typically multifactorial, for a model to generalize well to new datasets, a larger sample size may be needed to improve the recall and precision of the model and to finer assessment of the features.

Thank you!